LEVEL II

12

LOCKHEED

MISSILES & SPACE COMPANY, INC. • SUNNYVALE, CALIFORNIA

LEVEL II

12

DIRECT TIME INTEGRATION METHODS IN NONLINEAR STRUCTURAL DYNAMICS

C. A. Felippa
K. C. Park

Applied Mechanics Laboratory
Lockheed Palo Alto Research Laboratory
Palo Alto, California 94304 USA

LMSC-D013361

To be presented at the International Conference on Finite Elements in Nonlinear Mechanics (FENOMECH 1978) Institut für Statik und Dynamik der Luft- und Raumfahrkonstruktionen (ISD), Universität Stuttgart, Stuttgart, West Germany, August 30-September 1, 1978.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: LMSC-D013361
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): DIRECT TIME INTEGRATION METHODS IN NONLINEAR STRUCTURAL DYNAMICS
5. TYPE OF REPORT & PERIOD COVERED: Technical Report
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): Felippa, C.A. and Park, K.C.
8. CONTRACT OR GRANT NUMBER(s): N00014-74-C-0355

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Lockheed Palo Alto Research Laboratories
Applied Mechanics Laboratory
3251 Hanover St. Palo Alto, CA 94304

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS:
Department of Navy, Office of Naval Research
Structural Mechanics Program (Code 474)
Arlington, VA 22217

12. REPORT DATE: August 1978
13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public please, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
To be presented at the International Conference on Finite Elements in Nonlinear Mechanics (FENOMECH 78), University of Stuttgart, Stuttgart, West Germany - 1 Sept., 1978

18. SUPPLEMENTARY NOTES

C.A. Felippa K.C. Park

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Nonlinear structural dynamics, Direct time integration methods, Structural analysis software, Program modularity

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper reviews some recent developments in direct time integration methods for nonlinear structural dynamics. The developments pertain to the use of linear multistep difference operators in conjunction with the pseudo-force approach. The paper is organized into three main sections. An introductory section provides an overview of the transient response analysis problem. A section on computational aspects deals with the organization of the numerical calculations; this material is largely based on a recent detailed study of

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT cont'd:

linear dynamic calculations (1-2). A section on integration methods highlights algorithmic aspects that impact the selection of integrator for nonlinear problems and discusses adaptive analysis features such as stepsize control and implicit matrix scaling techniques. An appendix section outlines the functional organization of modular 'integration driving' software.

This paper is respectfully dedicated to Professor J. H. Argyris in honor of his 65th birthday and to Professor W. Prager in honor of his 75th birthday.

ABSTRACT

This paper reviews some recent developments in direct time integration methods for nonlinear structural dynamics. The developments pertain to the use of linear multistep difference operators in conjunction with the pseudo-force approach. The paper is organized into three main sections. An introductory section provides an overview of the transient response analysis problem. A section on computational aspects deals with the organization of the numerical calculations; this material is largely based on a recent detailed study of linear dynamic calculations [1-2]. A section on integration methods highlights algorithmic aspects that impact the selection of integrator for nonlinear problems and discusses adaptive analysis features such as stepsize control and implicit matrix scaling techniques. An appendix section outlines the functional organization of modular "integration driving" software.

Section 1
INTRODUCTION

1.1 The State of Computational Nonlinear Mechanics

Nonlinear dynamics is structural mechanics' new frontier. Although significant progress in the theory and implementation of computational methods has been made, nonlinear dynamics is presently an unsettled field and is likely to remain so in the near future.

A decade ago over 90% of engineering design work and perhaps over half of verification work was carried out using "linear thinking" mode. Of the nonlinear analysis fraction, an overwhelming proportion was devoted to static problems such as, for example, stability assessment. Applications of transient nonlinear analysis were hampered by three factors:

(1) Lack of reliable, computationally-oriented formulations;
(2) High computational costs;
(3) Unfamiliarity of prospective users with nonlinear mechanics in general and the use of application programs in particular.

Since then, these roadblocks have eroded in different degrees. Most dramatic progress has been made in (1). Obstacle (2) has weakened by gradual but steady decrease of unit computational costs as well as by improving efficiency of new implementations. The "education gap" (3) remains, however, a formidable barrier. Although it is easy to say that any problem in structural mechanics can be attacked from a nonlinear dynamics standpoint, in practice a converse route is followed. But it is often difficult for the nonspecialist to judge in advance whether nonlinearities and dynamic effects are of sufficient importance to merit consideration at some stage of the design or verification cycle. The main purpose of this section is to provide a quick overview of structural problems that may require a nonlinear dynamics treatment.

1.2 Motivation for Nonlinear Analysis

The increasing importance of nonlinear analysis is to a large extent due to emphasis placed by manufacturers, contractors and certifying agencies on realistic modeling and accurate analysis of critical structural components. This endeavor has prompted not only the incorporation of nonlinear analysis capabilities into existing linear codes, but also the development of new finite-element and finite-difference codes specifically designed for nonlinear analysis.

The need for nonlinear analysis capabilities was initially felt in the aerospace industry. Pressing demands for lightweight design led to the investigation of new structural configuration concepts (e.g., thin shear-carrying webs), new fabrication concepts (e.g., honeycomb panels), and increasing use of brittle materials such as fiber-reinforced composites. Implementation of these advanced concepts prompted concern over realistic modeling of structural components as regards numerical simulation of their behavior under critical loading and environmental conditions.

Continuing progress in variational discretization and solution techniques, coupled with the dissemination of production-level software, has helped in spreading interest in nonlinear structural analysis to non-aerospace applications occurring in automotive, bio, civil, industrial, nuclear and petroleum engineering. Among those systems that are often subject to significant nonlinear effects, we can mention thin shells, pressure vessels, nuclear reactors, cable and pneumatic structures, impacting structures, equipment mounts, shock-excited underwater and underground structures, and metal-forming machinery.

1.3 Static Versus Dynamic Treatment

As noted previously, an overwhelming proportion of engineering applications of nonlinear structural analysis are performed in "static mode" despite the undeniable presence of dynamic effects in many situations, such as snap-through analysis. This practice is commonly justified on various grounds.

First, it can be argued that many of the problem characterization parameters such as excitation and constitutive behavior are presently poorly understood and qualitatively uncertain to begin with. Second, the neglect of inertial effects often leads to conservative results. Third, dynamic analysis is both more cumbersome to carry out and more expensive than static analysis.

As more experience is gathered, the third argument weakens. In fact, nonlinear dynamic analysis is more easily implemented in a general purpose program framework than nonlinear static analysis. (This does not apply, of course, to linear analysis.) The only practical solution procedure for the former is direct time integration, whereas for the latter there is a vast array of special techniques that must be mastered by potential users. If computational costs continue to decrease in relation to manpower costs, it is not farfetched to presage that nonlinear static solution procedures could be eventually embedded into a dynamic analysis driver. If this occurs, the second argument becomes irrelevant. The first argument (poor problem characterization) remains valid, but an answer to this question lies primarily within the scope of experimental mechanics.

1.4 Sources of Nonlinearities

Problems in structural mechanics are formulated in terms of three spatial fields: displacements, strains, and stresses. These fields are connected by the stress-displacement and stress-strain (constitutive) relations as shown in Figure 1, which is adapted from [3]. The boundary value problem is closed by specification of the boundary conditions (B.C.) on displacements and/or stresses (or stress resultants). In dynamic analysis, the initial value problem is completed by specification of the initial displacements and velocities at a reference state.

Any of the four two-way links shown in Figure 1 may be nonlinear. Consequently, sources of nonlinearity can be naturally classified into four groups: force, material, geometric, and kinematic, which correspond to links 1, 2, 3, and 4, respectively.

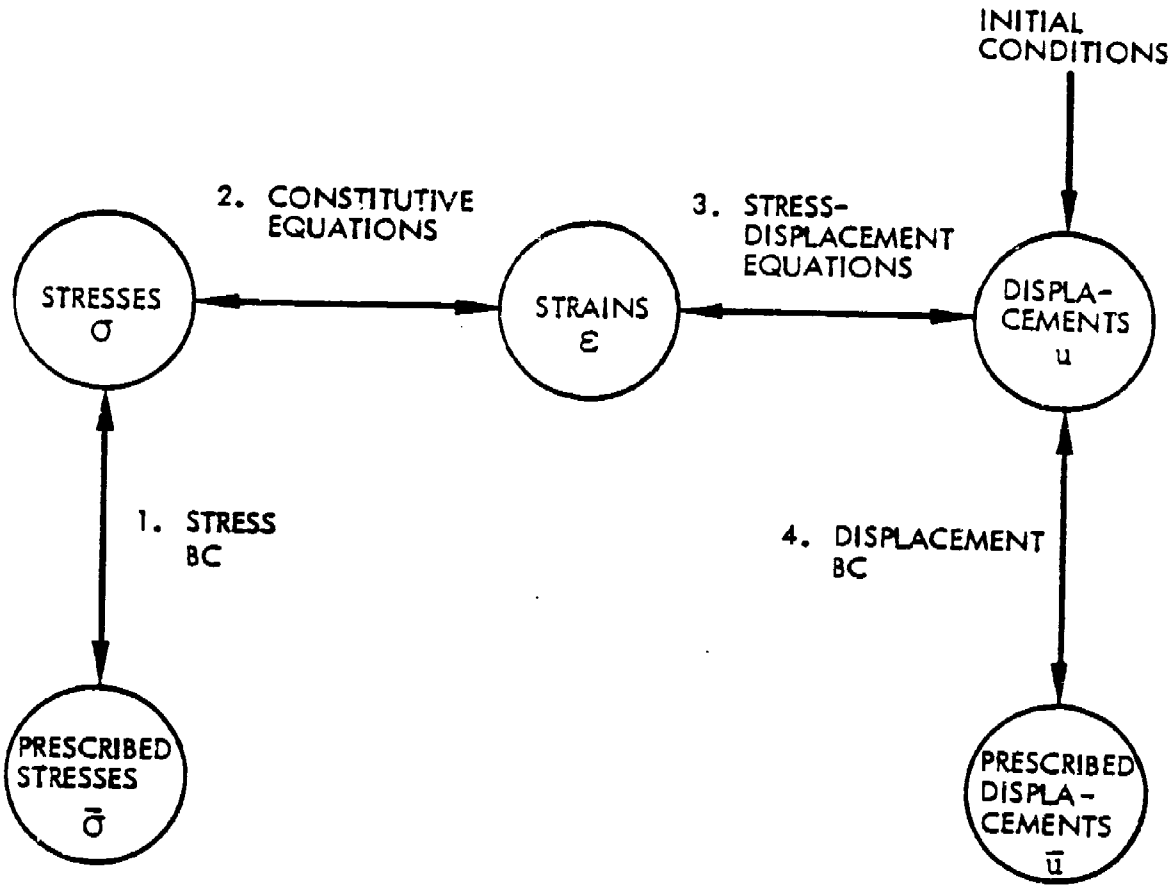


Fig. 1 Structural Analysis Fields and Connecting Relationships (after [3])

1.5 Governing Equations of Motion

The response of the physical structure to external excitation is approximated by that of a discrete mathematical model with n_f degrees of freedom, which are collected in a state vector $\underline{u}$. The components of $\underline{u}$ are generalized displacement coordinates that define the spatial configuration of the discrete model with respect to a reference state $\underline{u}_o$ for every time t. (This reference state may be updated during the solution of highly nonlinear problems.)

A contiguous family of configurations $\underline{u} = \underline{u}(t)$ satisfying initial and prescribed displacement conditions is called a motion. The governing equations of motion appropriate to the description of a wide range of nonlinear dynamic problems can be presented as follows [3]:

$$\underset{\sim}{M}\,\ddot{\underline{u}} + \underset{\sim}{D}\,\dot{\underline{u}} + \underline{C}(\underline{u},\dot{\underline{u}}) + \underset{\sim}{K}\,\underline{u} + \underline{S}(\underline{u}) = \underline{f}(t) \qquad (1.1)$$

where a superposed dot denotes temporal differentiation, and

- $\underset{\sim}{M}, \underset{\sim}{D}, \underset{\sim}{K}$ — mass, reference-damping, and reference-stiffness matrices, respectively (the "reference" concept is discussed in Section 1.6);
- $\underline{C}, \underline{S}$ — nonlinear damping and stiffness operators, respectively, which generate state-dependent corrective force vectors;
- $\underline{f}(t)$ — vector of applied (generalized) forces.

The motion $\underline{u}(t)$ corresponding to a given $\underline{f}(t)$ will be called the response of the system (1.1). Essential B.C. on specific components of $\underline{u}(t)$ are assumed to be embodied in (1.1) for notational convenience. The effect of non-homogeneous essential B.C. can be also incorporated in (1.1) through various techniques such as the adjunction of penalty terms or the introduction of Lagrange multipliers.

The generation of system (1.1) from the continuous field model schematized in Figure 1 will not be dealt with in this paper, which is concerned with techniques for solving it.

Remark 1. Equation (1.1) does not cover the esoteric class of mass-varying problems such as reentry vehicle dynamics.

Remark 2. Equation (1.1) as written does not model "path dependent" problems, in which the response is a functional of the past motion. These problems can be accounted for by making the nonlinear terms memory-functional operators.

Remark 3. In explicit time integration, $K = 0$, and all stiffness information is transmitted in vector form.

1.6 The Pseudo-Force Approach

Implicit direct time integration procedures for solving (1.1) can be based on either the tangent stiffness or a pseudo-force approach. In the former, which is briefly discussed in Section 2.8, (1.1) is linearized at each computed state and the solution advanced incrementally. In the latter, the reference matrices $\underset{\sim}{K}$ and $\underset{\sim}{D}$ are kept fixed over many computational steps and the nonlinear terms $\underline{S}$ and $\underline{C}$ are treated as state-dependent reaction forces; this is the approach emphasized herein.

The application of the pseudo-force approach to transient nonlinear analysis was first advocated by Stricklin, Haisler and coworkers [4]. Their strategy was to use the linear matrices $\underset{\sim}{K}$ and $\underset{\sim}{D}$ evaluated at the initial configuration. We can, however, do much better from a global point of view if the selection of $\underset{\sim}{K}$ and $\underset{\sim}{D}$ is not so restricted; in fact, such matrices need not be identifiable with the stiffness and damping matrices evaluated at actual configurations. This generalized pseudo-force method, presented by Underwood and Park [5], is designed to work effectively in conjunction with the matrix scaling technique described in Section 3.

The generalized stiffness selection concept may be illustrated with the bilinear spring shown in Figure 2. In the conventional pseudo-force approach, the stiffness would be based on the initial slope k_o. In the approach described here, an average slope $\bar{k}$ can be used to evenly distribute the force residual magnitude over the expected operational range (note that $\bar{k}$ is not physically attainable).

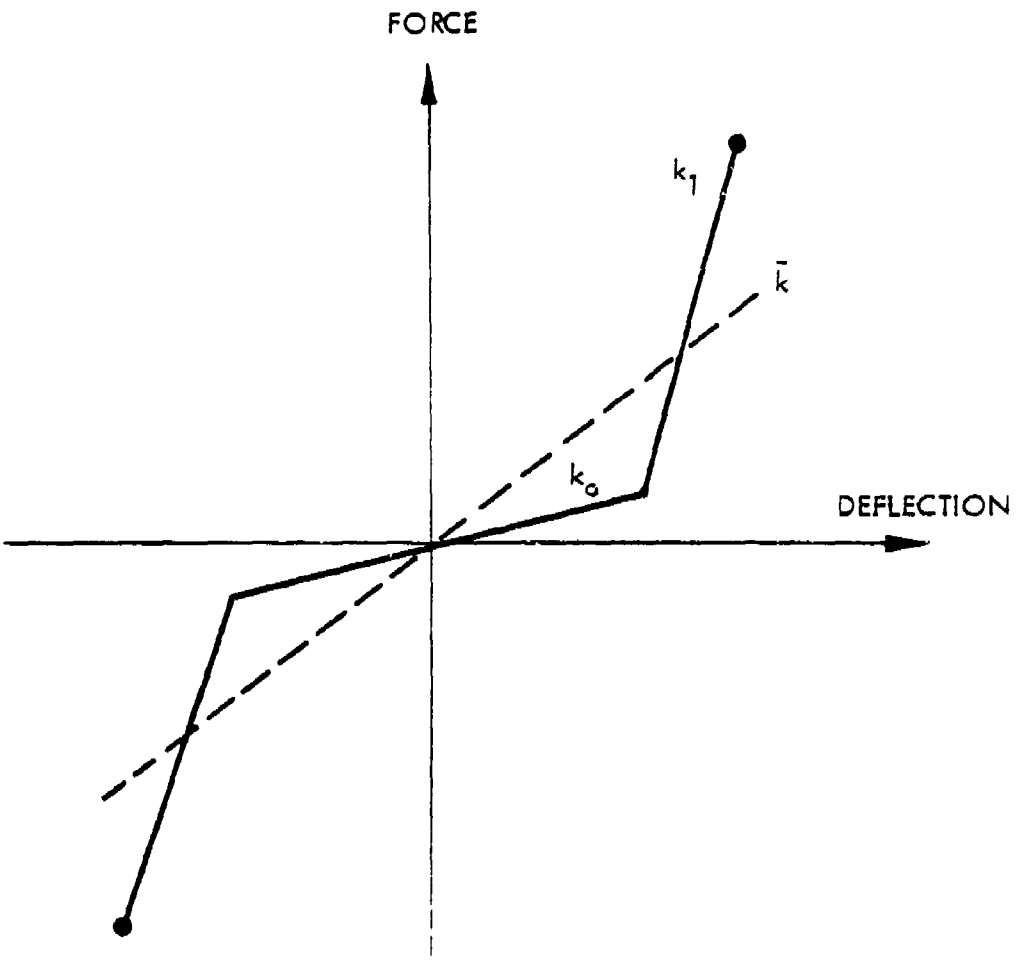


Fig. 2 Reference Stiffness Selection for Bilinear Spring Element (after [5])

A helpful guideline is: integration procedures based on the pseudo-force approach experience less difficulties with "softening" systems (e.g., elasto-plastic materials) than with "hardening" systems (e.g., tensile structures). It is therefore best to err on the side of overestimating the initial reference stiffness. As a simple application of this rule, consider the impact analysis of two structures, S_1 and S_2, which are initially separated. Then the reference stiffness should be that of S_1+S_2 moving together.

1.7 Classification of Nonlinear Dynamics Problems

A simple classification scheme can be based on the effect of the spectral characteristics of the excitation on the overall structural response:

> Structural Dynamics (SD) Problems: The response is controlled by a relatively small number of low-frequency modes.
>
> Wave Propagation (WP) Problems: The contribution of intermediate and high-frequency structural modes to the response remains important throughout the time span of interest.

In the above, the terms "frequencies" and "modes" refer to the eigenspectrum of the linearization of the left-hand side of (-1); because of nonlinearities this spectrum can be expected to vary (usually mildly) with the state. The qualifiers low, intermediate, and high refer to frequencies whose associated wavelengths are much larger than, of the order of, and much smaller than the characteristic acoustic wavelengths, respectively.

In practice, a combination of the preceding types is often encountered:

> Shock-excited Problems: An initial high-frequency transient gradually decays to a steady state or free vibration regime (step waves, earthquakes, single impact, separation phenomena, accidents).
>
> Multishock-excited Programs: Multiple high-frequency transients occur with some regularity, possibly leading to gradual collapse or fatigue phenomena (reflecting shock waves, repeated impacts, periodic deployment).

Dynamic Instability: Relatively smooth, bounded response suddenly "takes off" as critical conditions are attained (follower forces, flutter, parametric excitation, phase changes).

Generally speaking, implicit integration methods are most effective for SD problems (or phases) while explicit integration methods are at their best for tracing WP problems (or phases).

1.8 Organization of the Paper

Direct time integration techniques for (1.1) can be studied from two standpoints:

1. Algorithmic: Study of mathematical properties such as stability and accuracy of the integration formulas.

2. Computational: Study of properties directly correlated with the implementation of the integration procedure on a computer.

An array of subjects pertaining to the latter viewpoint is presented in Figure 3 in logical flowchart format. The basic formulation steps leading to the "advancing cycle" box (central column in Figure 3) are covered in Section 2.

Section 3 reviews algorithmic aspects that impact the selection of integration formulas for nonlinear dynamics, and discusses features labeled as "advanced" in Figure 3, namely accuracy control via automatic stepsize selection, and the matrix scaling technique.

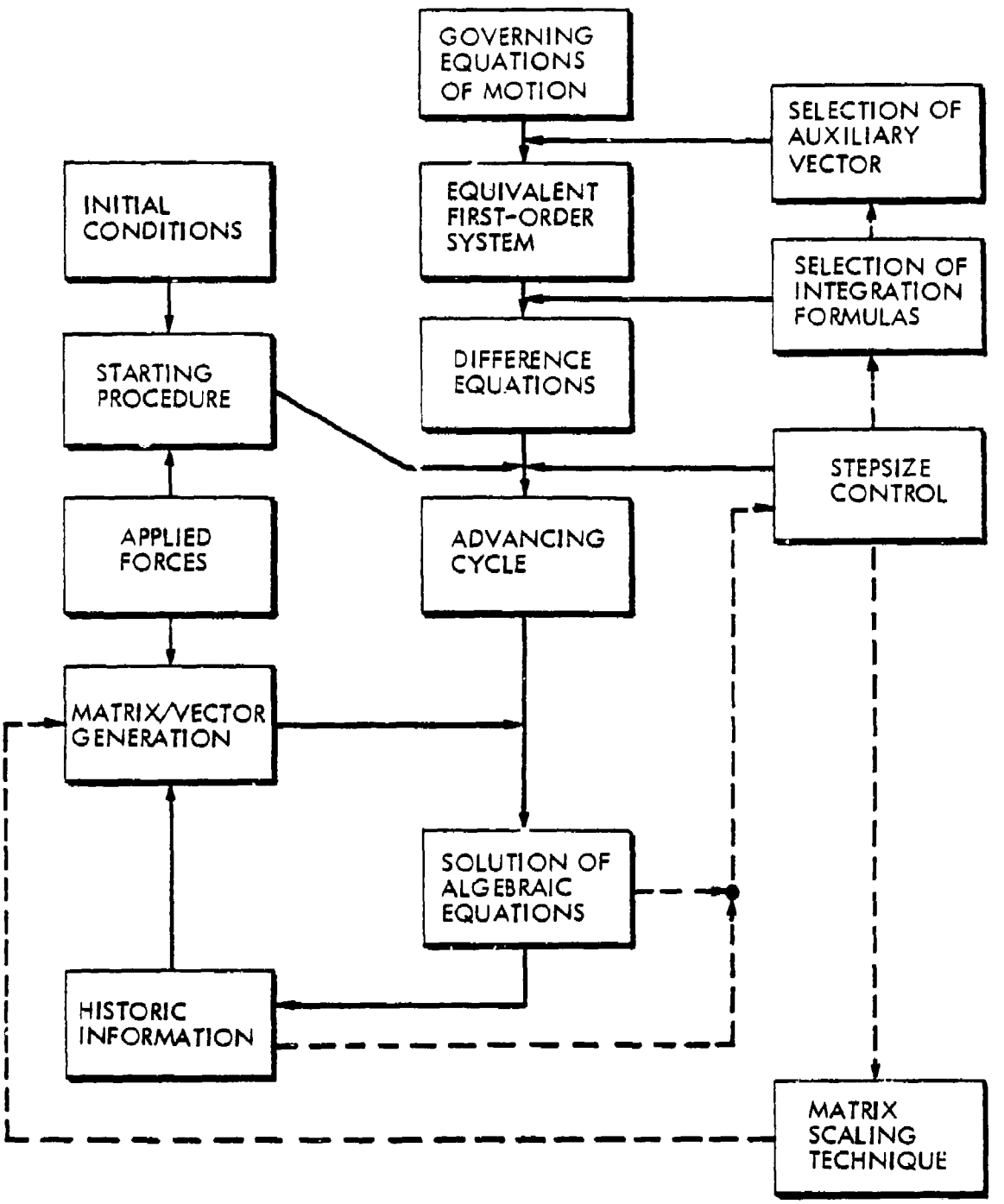


Fig. 3 Logical Flow Diagram of Direct Time Integration Process (---→: Advanced Features)

Section 2

COMPUTATIONAL ASPECTS

2.1 Reduction to a First-Order System

Equation (1.1) can be reduced to a first order system by introducing the auxiliary vector $\underline{v}$ [6]:

$$\underline{v} = \underset{\sim}{A}\,\underset{\sim}{M}\,\dot{\underline{u}} + \underset{\sim}{B}\,\underline{u} \tag{2.1}$$

$$\begin{aligned}\dot{\underline{v}} &= \underset{\sim}{A}\,\underset{\sim}{M}\,\ddot{\underline{u}} + \underset{\sim}{B}\,\dot{\underline{u}} \\ &= \underset{\sim}{A}\,(\underline{f} - \underline{C} - \underline{S} - \underset{\sim}{K}\,\underline{u} - \underset{\sim}{D}\,\dot{\underline{u}}) + \underset{\sim}{B}\,\dot{\underline{u}}\end{aligned} \tag{2.2}$$

where $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are (for the moment) arbitrary n_f by n_f matrices with $\underset{\sim}{A}$ non-singular. These matrices are independent of time and state, although they can be functions of the integration stepsize. The resulting first-order system (2.1 - 2.2) can be written

$$\begin{bmatrix} \underset{\sim}{A}\,\underset{\sim}{M} & \underset{\sim}{Q} \\ \underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B} & \underset{\sim}{I} \end{bmatrix} \begin{Bmatrix} \dot{\underline{u}} \\ \dot{\underline{v}} \end{Bmatrix} + \begin{bmatrix} \underset{\sim}{B} & -\underset{\sim}{I} \\ \underset{\sim}{A}\,\underset{\sim}{K} & \underset{\sim}{Q} \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{v} \end{Bmatrix} = \begin{Bmatrix} \underline{Q} \\ \underset{\sim}{A}\,\underline{Q} \end{Bmatrix} \tag{2.3}$$

where $\underset{\sim}{I}$ and $\underset{\sim}{Q}$ denote the identity and null matrix, respectively, and $\underline{Q}$ is the effective force

$$\underline{Q} = \underline{f} - \underline{C} - \underline{S} \tag{2.4}$$

Note that the nonlinear terms $\underline{C}$ and $\underline{S}$ have been transferred to the right-hand side as "pseudo-forces." Equation (2.3) fits within the standard O.D.E. format

$$\dot{\underline{y}} = \underline{F}(\underline{y},t) \tag{2.5}$$

where $\underline{y}^t = (\underline{u}^t \; \underline{v}^t)$, and

$$\underline{F} = \begin{bmatrix} \underset{\sim}{A}\,\underset{\sim}{M} & \underset{\sim}{0} \\ \underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B} & \underset{\sim}{I} \end{bmatrix}^{-1} \begin{Bmatrix} \underline{v} - \underset{\sim}{B}\,\underline{u} \\ \underset{\sim}{A}\,(\underline{Q} - \underset{\sim}{K}\,\underline{u}) \end{Bmatrix} \tag{2.6}$$

2.2 Time Discretization by LMS Formulas

We consider the numerical integration of the first-order system (2.3) using m-step, one-derivative, linear multistep (LMS) difference operators to discretize the time variation of $\underline{u}$ and $\underline{v}$. A detailed treatment of these methods can be found elsewhere, e.g.,[7-9]. For a constant stepsize h, LMS operators applied to Eq.(2.3) at the n-th time station t_n can be written

$$\left.\begin{aligned} \sum_{i=0}^{m} \alpha_i \, \underline{u}_{n-i} &= h \sum_{i=0}^{m} \beta_i \, \dot{\underline{u}}_{n-i} \\ \sum_{i=0}^{m} \gamma_i \, \underline{v}_{n-i} &= h \sum_{i=0}^{m} \delta_i \, \dot{\underline{v}}_{n-i} \end{aligned}\right\} \quad \begin{aligned} &m \geq 1, \; n \geq m \\ &\alpha_o \neq 0, \; \gamma_o \neq 0 \end{aligned} \tag{2.7}$$

where α_i, β_i, γ_i, and δ_i are scalars associated with specific finite difference operators, and $\underline{u}_k$, $\underline{v}_k$, ... stand for the vectors $\underline{u}(t_k)$, $\underline{v}(t_k)$, ... , computed through the discretization procedure (2.7). The above difference expressions may be scaled by arbitrary factors. In the sequel they will be normalized by requiring that

$$\alpha_o = \gamma_o = 1 \tag{2.8}$$

If β_o is nonzero (zero), the operator (2.7a) is said to be implicit (explicit), and similarly for (2.7b).

With a view to subsequent manipulations, it is convenient to introduce a more compact notation which, nevertheless, separates explicitly the current

state variables $\underline{u}_n$, $\dot{\underline{u}}_n$, $\underline{v}_n$, $\dot{\underline{v}}_n$ from historical terms involving past solutions:

$$\begin{aligned} \underline{u}_n + \underline{a}^u_n &= h_\beta \dot{\underline{u}}_n + h\, \underline{b}^{\dot{u}}_n \\ \underline{v}_n + \underline{c}^v_n &= h_\delta \dot{\underline{v}}_n + h\, \underline{d}^{\dot{v}}_n \end{aligned} \tag{2.9}$$

where the "generalized stepsizes" h_β, h_δ are defined by

$$h_\beta = h\,\beta_o \qquad h_\delta = h\,\delta_o \tag{2.10}$$

and vectors $\underline{a}$, $\underline{b}$, ... denote linear combinations of past solutions:

$$\left[\underline{a}^z_n\ \underline{b}^z_n\ \underline{c}^z_n\ \underline{d}^z_n\right] = \left[\underline{z}_{n-1} \cdots \underline{z}_{n-m}\right] \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & \delta_1 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_m & \beta_m & \gamma_m & \delta_m \end{bmatrix} \tag{2.11}$$

In Eq. (2.11), $\underline{z}$ is an arbitrary vector symbol, which may stand for $\underline{u}$, $\dot{\underline{u}}$, ... etc. The current state solution $(\underline{u}_n, \underline{v}_n)$ can be expressed from (2.9) in the fo l

$$\begin{Bmatrix} \underline{u}_n \\ \underline{v}_n \end{Bmatrix} = \begin{Bmatrix} h_\beta \dot{\underline{u}}_n \\ h_\delta \dot{\underline{v}}_n \end{Bmatrix} + \begin{Bmatrix} \underline{h}^u_n \\ \underline{h}^v_n \end{Bmatrix} \tag{2.12}$$

where

$$\begin{aligned} \underline{h}^u_n &= h\, \underline{b}^{\dot{u}}_n - \underline{a}^u_n \\ \underline{h}^v_n &= h\, \underline{d}^{\dot{v}}_n - \underline{c}^v_n \end{aligned} \tag{2.13}$$

will be called, following Jensen [6], the historical vectors pertaining to $\underline{u}$ and $\underline{v}$, respectively.

2.3 Implicit Integration

We consider first implicit LMS operators, for which both β_o and δ_o are nonzero, whence $h_\beta \neq 0$ and $h_\delta \neq 0$. Elimination of $\dot{\underline{u}}_n$ and $\dot{\underline{v}}_n$ from (2.3) and (2.13) then produces the nonlinear algebraic system of order $2\,n_f$:

$$\begin{bmatrix} \underset{\sim}{A}\,\underset{\sim}{M} + h_\beta\,\underset{\sim}{B} & -h_\beta\,\underset{\sim}{I} \\ \underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B} + h_\beta\,\underset{\sim}{A}\,\underset{\sim}{K} & \eta\,\underset{\sim}{I} \end{bmatrix} \begin{Bmatrix} \underline{u}_n \\ \underline{v}_n \end{Bmatrix} = \begin{Bmatrix} \underset{\sim}{A}\,\underset{\sim}{M}\,\underline{h}_n^u \\ (\underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B})\,\underline{h}_n^u + \eta \underline{h}_n^v + h_\beta\,\underset{\sim}{A}\,\underline{Q}_n \end{Bmatrix} \tag{2.14}$$

where $\eta = h_\beta/h_\delta = \beta_o/\delta_o$, and $\underline{Q}_n = \underline{f}_n - \underline{C}_n - \underline{S}_n$. Finally, elimination of $\underline{v}_n$ from (2.14) yields the n_f-th order nonlinear algebraic system

$$\underset{\sim}{E}\,\underline{u}_n = \underline{q}_n \tag{2.15}$$

where

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{M} + h_\delta\,\underset{\sim}{D} + h_\beta\,h_\delta\,\underset{\sim}{K} + (h_\beta - h_\delta)\,\underset{\sim}{G} \\ \underline{q}_n &= \left[\underset{\sim}{M} + h_\delta(\underset{\sim}{D}-\underset{\sim}{G})\right] \underline{h}_n^u + h_\beta\,\underset{\sim}{A}^{-1}\,\underline{h}_n^v + h_\beta\,h_\delta\,\underline{Q}_n \\ \underset{\sim}{G} &= \underset{\sim}{A}^{-1}\,\underset{\sim}{B} \end{aligned} \tag{2.16}$$

The selection of a common LMS operator for both $\underline{u}$ and $\underline{v}$ is natural if the analyst wishes to keep similar discretization accuracy on both vectors. In this case, Eqs. (2.9) become

$$\begin{aligned} \underline{u}_n + \underline{a}_n^u &= h_\beta\,\dot{\underline{u}}_n + h\,\underline{b}_n^{\dot{u}} \\ \underline{v}_n + \underline{a}_n^v &= h_\beta\,\dot{\underline{v}}_n + h\,\underline{b}_n^{\dot{v}} \end{aligned} \tag{2.17}$$

and the components of (2.15) simplify slightly to

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{M} + h_\beta\,\underset{\sim}{D} + h_\beta^{\,2}\,\underset{\sim}{K} \\ \underline{q}_n &= \left[\underset{\sim}{M} + h_\beta\,(\underset{\sim}{D} - \underset{\sim}{G})\right] \underline{h}_n^u + h_\beta\,\underset{\sim}{A}^{-1}\,\underline{h}_n^v + h_\beta^{\,2}\,\underline{Q}_n \end{aligned} \tag{2.18}$$

The coefficient matrix $\underset{\sim}{E}$ becomes independent of the choice of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in this case, as noted by Jensen [6].

Remark 1. The generality of the discretization (2.7), in which different LMS operators may be used for $\underline{u}$ and $\underline{v}$, has two important advantages. First, the use of different approximations for $\underline{u}$ and $\underline{v}$ is allowed, should that prove desirable on account of the physical meaning of $\underline{v}$ resulting from specific choices of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in (2.2). Second, special two-derivative integrators devised for treating the second order system (1.1) directly, such as the widely used Newmark [10], Houbolt [11], and Wilson [12] operators, can be presented as special cases of a slight extension of (2.7), in which a historical $\underline{u}$-term is appended to (2.7b), as shown in Reference [1]. It follows that a separate study of the computer implementation of these methods is not required.

Remark 2. If $h_\beta \neq h_\delta$, the choice of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ should be such that $\underset{\sim}{G} = \underset{\sim}{A}^{-1}\,\underset{\sim}{B}$ is symmetric and maintains the sparseness characteristics of the structural matrices. (All of the choices discussed in Section 2.5 comply with these restrictions.)

Remark 3. If two different A-stable operators are used for $\underline{u}$ and $\underline{v}$, and $\underset{\sim}{B}$ is nonzero, the resulting system (2.14) is not necessarily A-stable. Stability conditions have to be checked out in advance to avoid surprises.

2.4 Computational Paths for Implicit Integration

Implementations of the implicit integration procedure outlined in the preceding section may differ in two respects: (a) the selection of weighting matrices $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in (2.1), and (b) the manner in which the resulting auxiliary vector $\underline{v}$ and its temporal derivative $\dot{\underline{v}}$ are computed at each step. We deal with the latter topic first.

Three basic computational paths, labeled (0), (1), and (2), may be followed in advancing the discrete solution over a typical time step. The associated sequences of calculations are flow charted in Figure 4.

The path identification index, (0-2), gives an idea of the number of backward

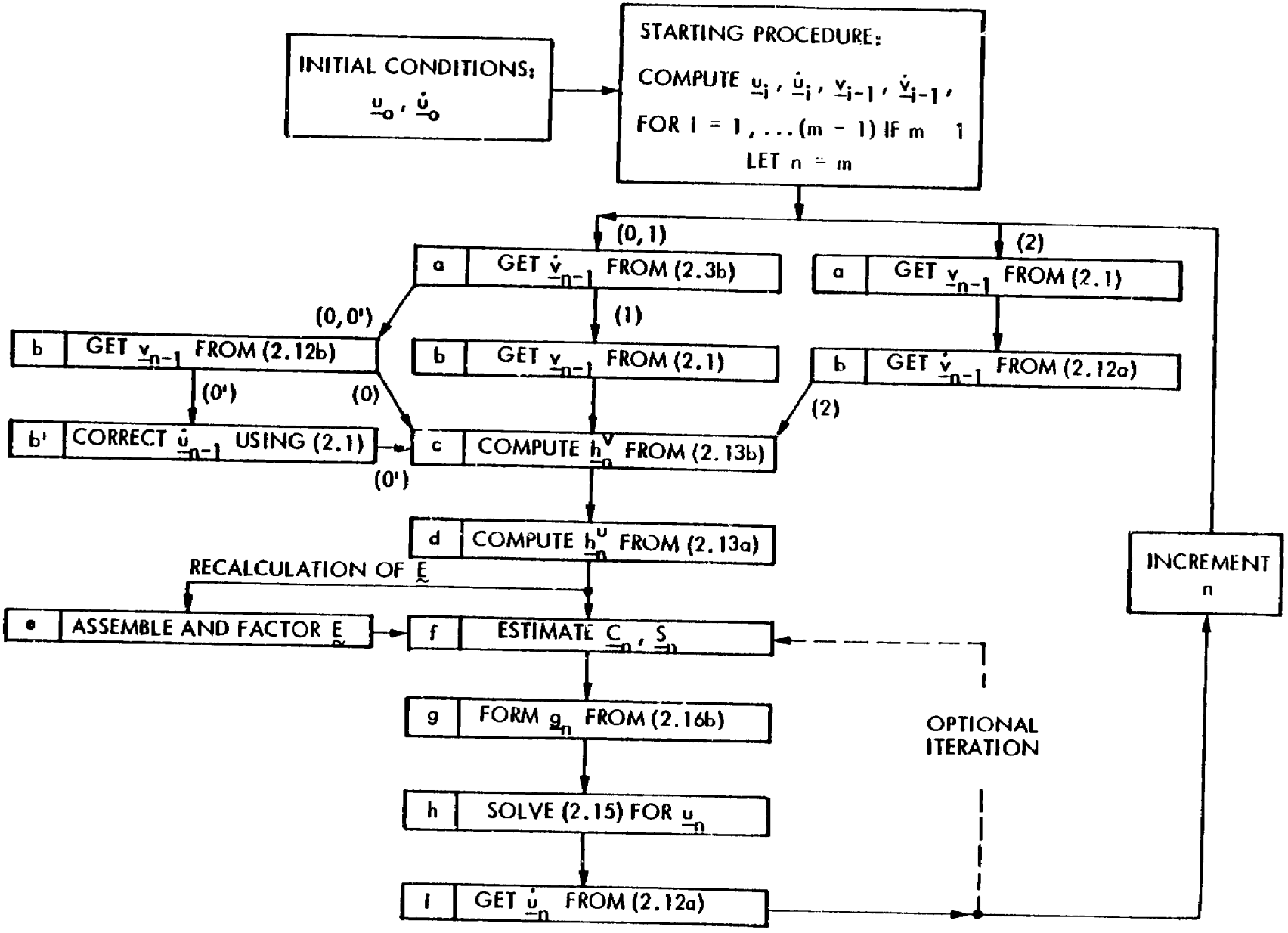


Fig. 4 Computational Paths for Pseudo-Force Formulation of Nonlinear Dynamic Analysis

difference operations performed in the determination of $\underline{v}_n$ and $\dot{\underline{v}}_n$ in each solution advancing cycle. It is shown in Reference [2] that this index plays an important role in the error propagation behavior of particular implementations.

Path (0) is consistent with the first-order difference system (2.14), i.e., the computed vectors satisfy this system exactly if computational errors are neglected and the solution of the nonlinear system (2.15) is iterated to convergence. (The solution error can actually be regarded as a component of the "local" computational error [2].) The original O.D.E. system (1.1) is also satisfied. On the other hand, the computed $\underline{u}_n$, $\dot{\underline{u}}_n$ and $\underline{v}_n$ do not, in general, verify the auxiliary vector definition (2.1), which is satisfied only in the limit $h \to 0$.

A slight variant of path (0), labeled (0'), is also shown in Figure 4. In this variant the velocity vector $\dot{\underline{u}}_{n-1}$ is recomputed so that (2.1) is verified at past stations. This "delayed correction" is not only expensive for general $\underset{\sim}{A}$ and $\underset{\sim}{B}$ but worthless as regards improvement of error propagation characteristics [1,2]. It is included in Figure 4, however, because it occurs naturally in the conventional choice $\underline{v} = \dot{\underline{u}}$ discussed in Section 2.5.

Path (1) satisfies both differential expressions (1.1) and (2.1) at the expense of (2.14b). Finally, path (2) enforces the $\underline{v}$ definition (2.1) and the difference system (2.14) at the expense of (1.1).

2.5 Selection of Auxiliary Vector

The computational effort per time step (and, to a lesser extent, the storage requirements) can be significantly reduced for certain choices of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ because some of the computational steps displayed in Figure 4 can be either simplified or entirely bypassed. The following two selections were studied in detail in Reference [1] for linear dynamic problems:

$$\underline{v} = \dot{\underline{u}} \qquad (\underset{\sim}{A} = \underset{\sim}{M}^{-1}\ ,\ \underset{\sim}{B} = \underset{\sim}{0}) \tag{2.19}$$

$$\underline{v} = \underset{\sim}{M}\,\dot{\underline{u}} + \underset{\sim}{D}\,\underline{u} \qquad (\underset{\sim}{A} = \underset{\sim}{I}\ ,\ \underset{\sim}{B} = \underset{\sim}{D}) \tag{2.20}$$

Equations (2.19) and (2.20) were labeled the conventional (C) and Jensen's (J) formulations, respectively. When these two choices are combined with the four computational paths of Figure 4, a total of eight formulations of the advancing step result. The comparative study carried out in Reference [1] identified the following combinations as preferable:

(C1) Minimizes computational effort for umdamped systems when used in conjunction with backward-difference operators and requires less vector storage than (J0), but does not possess favorable error propagation control properties.

(J0) Minimizes computational effort per step for generally damped systems and has excellent error propagation control properties, but requires additional vector storage.

In both these formulations, the presence of an ill-conditioned or singular mass matrix $\underset{\sim}{M}$ does not cause any particular difficulty. There are also no problems associated with the starting procedure. Similar conclusions can be extended to transient nonlinear analysis handled with the pseudo-force approach. The optional corrective iteration cycle shown in Figure 4 does not affect computational effort rankings provided invariant calculations are moved out of the loop. The computational sequences associated with the (C1) and (J0) formulations are shown in Tables 1 and 2, respectively.

Additional formulations of interest in the nonlinear case are obtained if matrices $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are allowed to be stepsize-dependent. For example, the selection

$$\underline{v} = \underset{\sim}{M}\,\dot{\underline{u}} + (\underset{\sim}{D} + h_\beta \underset{\sim}{K})\,\underline{u} \tag{2.21}$$

in conjunction with path (0) leads to a variant of the (J0) formulation that will be denoted by (J0K). The associated computational sequence is shown in Table 3. Its more interesting feature is the relocation of steps (a) and (b), which is believed to offer some advantages in array management if an iteration loop is applied.

Table 1

COMPUTATIONAL SEQUENCE ASSOCIATED WITH THE (C1) FORMULATION

Step	Calculation
a	$\underset{\sim}{M}\,\ddot{\underline{u}}_{n-1} = \underline{f}_{n-1} - \underset{\sim}{D}\,\dot{\underline{u}}_{n-1} - \underset{\sim}{K}\,\underline{u}_{n-1}$
b	$\dot{\underline{u}}_{n-1} = \dot{\underline{u}}_{n-1}$ (trivial)
c	$\underset{\sim}{M}\,\dot{\underline{h}}^{u} = h\,\underline{d}_n^{M}\,\ddot{u} - \underset{\sim}{M}\,\underline{c}_n^{\dot{u}}$
d	$\underline{h}_n^{u} = h\,\underline{b}_n^{\dot{u}} - \underline{a}_n^{u}$
e[1]	$\underset{\sim}{E} = \underset{\sim}{M} + h_\delta\,\underset{\sim}{D} + h_\beta\,h_\delta\,\underset{\sim}{K}$
f	Predict or re-evaluate $\underline{C}_n$, $\underline{S}_n$ ← - - - - - - - -
g	$\underline{q}_n = (\underset{\sim}{M} + h_\delta\,\underset{\sim}{D})\,\underline{h}_n^{u} + h_\beta\,\underset{\sim}{M}\,\dot{\underline{h}}_n^{u} + h_\beta\,h_\delta\,\underline{Q}_n$
h	$\underline{u}_n = \underset{\sim}{E}^{-1}\,\underline{q}_n$
i	$\dot{\underline{u}}_n = (\underline{u}_n - \underline{h}_n^{u})/h_\beta$ Optional Iteration[2] - - - - - - - - - - - - - -

NOTES:

(1) $\underset{\sim}{E}$ is assembled and factored only occasionally, cf. Section 3.

(2) If iteration is used, calculation of the invariant portion of $\underline{q}_n$ should be moved outside the loop.

Table 2

COMPUTATIONAL SEQUENCE ASSOCIATED WITH THE (JO) FORMULATION

Step	Calculation
a	$\dot{\underline{v}}_{n-1} = \underline{Q}_{n-1} - \underset{\sim}{K}\,\underline{u}_{n-1}$
b	$\underline{v}_{n-1} = \underline{h}^{v}_{n-1} + h_{\delta}\,\dot{\underline{v}}_{n-1}$
c	$\underline{h}^{v}_{n} = h\,\underline{d}^{\dot{v}}_{n} - \underline{c}^{v}_{n}$
d	$\underline{h}^{u}_{n} = h\,\underline{b}^{\dot{u}}_{n} - \underline{a}^{u}_{n}$
e(1)	$\underset{\sim}{E} = \underset{\sim}{M} + h_{\beta}\,\underset{\sim}{D} + h_{\delta}\,h_{\beta}\underset{\sim}{K}$
f	Predict or re-evaluate $\underline{C}_n$, $\underline{S}_n$
g	$\underline{g}_n = \underset{\sim}{M}\,\underline{h}^{u}_{n} + h_{\beta}\,\underline{h}^{v}_{n} + h_{\delta}\,h_{\beta}\,\underline{Q}_n$
h	$\underline{u}_n = \underset{\sim}{E}^{-1}\,\underline{g}_n$
i	$\dot{\underline{u}}_n = (\underline{u}_n - \underline{h}^{u}_{n})/h_{\beta}$ — Optional Iteration(2)

NOTES: (1), (2), see Table 1.

Table 3

COMPUTATIONAL SEQUENCE ASSOCIATED WITH THE (J0K) FORMULATION

Step	Calculation
c	$\underline{h}_n^v = h\,\underline{d}_n^v - \underline{c}_n^v$
d	$\underline{h}_n^u = h\,\underline{b}_n^u - \underline{a}_n^u$
e(1)	$\underline{\underline{E}} = \underline{\underline{M}} + h_\beta \underline{\underline{D}} + h_\beta^2 \underline{\underline{K}}$
f	Predict or re-evaluate $\underline{C}_n$, $\underline{S}_n$
a	$\dot{\underline{v}}_n = \underline{Q}_n - \underline{\underline{K}}\,\underline{h}_n^u$
g	$\underline{q}_n = \underline{\underline{M}}\,\underline{h}_n^u + h_\beta\,\underline{h}_n^v + h_\beta^2\,\dot{\underline{v}}_n$
h	$\underline{u}_n = \underline{\underline{E}}^{-1}\,\underline{q}_n$
i	$\dot{\underline{u}}_n = (\underline{u}_n - \underline{h}_n^u)/h_\beta$ — Optional Iteration(2)
b	$\underline{v}_n = \underline{h}_n^v + h_\delta\,\dot{\underline{v}}_n$

NOTES: (1), (2) see Table 1.

Similarly, the use of the auxiliary vector

$$\underset{\sim}{v} = \underset{\sim}{M} (\underline{u}_n - h_\beta \dot{\underline{u}}_n) = \underset{\sim}{M} \underline{h}_n^u \tag{2.22}$$

in conjunction with path (2) allows a similar relocation of steps (a) through (b); its practical value has not been assessed, however.

Table 4 provides a "snapshot" view of vector structures involved in the programming of the (J0K) sequence. The flow of computations required to advance the solution over the n-th step is indicated with arrowed paths. (Similar diagrams for the (C1) and (J0) sequences can be found in [1].)

2.6 Primary and Secondary Computational Variables

In the formulations discussed in previous sections, the displacement vector $\underline{u}_n$, which is obtained from solving (2.15), is called the primary computational variable; the other three state vectors ($\dot{\underline{u}}_n$, $\underline{v}_n$, $\dot{\underline{v}}_n$) are secondary or derived. We could have solved (2.3) and (2.12) for another state vector. For example, the counterpart of (2.15) for the velocities is

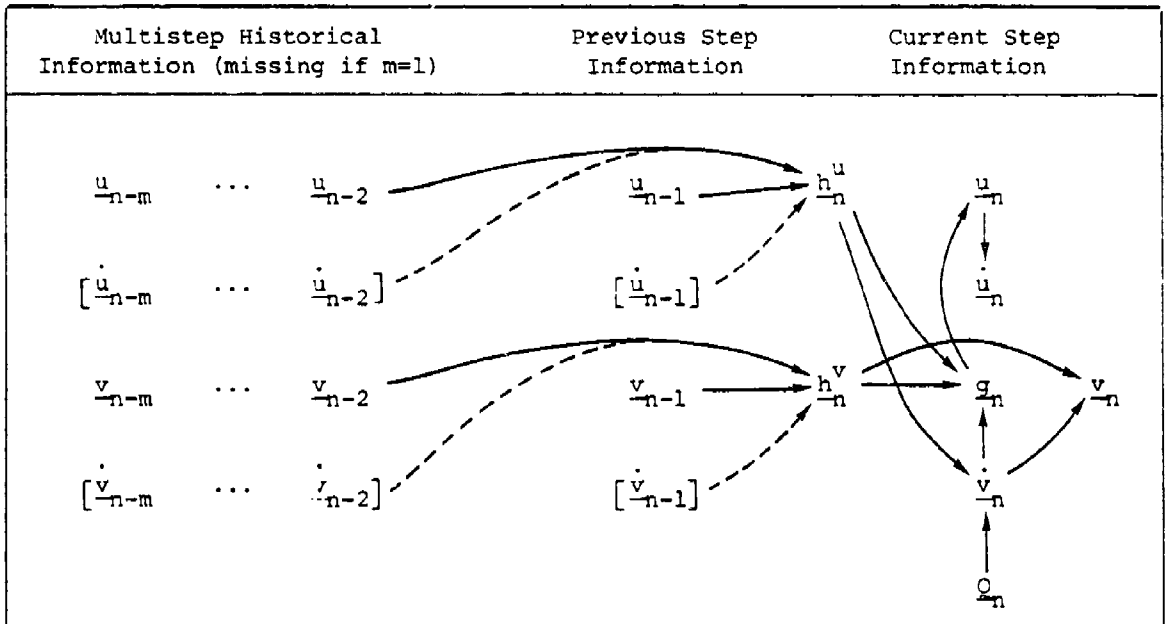
$$\underset{\sim}{E}\, \dot{\underline{u}}_n = - (h_\delta \underset{\sim}{K} + \underset{\sim}{G})\, \underline{h}_n^u + \underset{\sim}{A}^{-1}\, \underline{h}_n^v + h_\delta \underline{Q}_n \tag{2.23}$$

The displacements $\underline{u}_n$ would then be computed from (2.12a); this sequence effectively reverses steps (h) and (i) in Figure 4. (Corresponding systems for $\underline{v}_n$ and $\dot{\underline{v}}_n$ are considerably messier for general $\underset{\sim}{A}$ and $\underset{\sim}{B}$.)

If exact arithmetic were used throughout, the selection of primary variable would be irrelevant. The choice does matter, however, if computational error propagation is considered. Generally speaking, the state variable of main interest in the analysis should be selected as primary. For structural dynamics problems, displacements are a natural choice. In fluid-structure interaction problems, however, structural velocities are of utmost concern and have been used as primary computational variables in the implementation of staggered solution procedures [13].

Table 4

VECTOR STORAGE ARRANGEMENT FOR (J0K) FORMULATION



NOTES:

1. If a pair of backward difference operators ($\underline{b} = \underline{d} = \underline{0}$) is used, information enclosed in brackets is not required for advancing the computations, and those computational steps identified by broken-line paths may be omitted.

2. If iterations are applied, the arrangement is slightly more complex, since invariant vector portions must be removed from the iteration loop.

2.7 Explicit Integration

If two explicit integration procedures ($\beta_o = \delta_o = 0$) are selected in (2.7), the computational sequence becomes

$$\begin{aligned} \underline{u}_n &= \underline{h}_n^u \\ \underline{v}_n &= \underline{h}_n^v \\ \dot{\underline{u}}_n &= \underset{\sim}{M}^{-1} \underset{\sim}{A}^{-1} (\underline{v}_n - \underset{\sim}{B}\, \underline{u}_n) \\ \dot{\underline{v}}_n &= \underset{\sim}{A} (\underline{Q}_n - \underset{\sim}{K}\, \underline{u}_n - \underset{\sim}{D}\, \dot{\underline{u}}_n) + \underset{\sim}{B}\, \dot{\underline{u}}_n \end{aligned} \tag{2.24}$$

Economic execution of this sequence demands the presence of a diagonal (and nonsingular) mass matrix $\underset{\sim}{M}$. In addition, the conventional formulation $\underline{v} = \dot{\underline{u}}$, $\dot{\underline{v}} = \ddot{\underline{u}}$ ($\underset{\sim}{A} = \underset{\sim}{M}^{-1}$, $\underset{\sim}{B} = \underset{\sim}{0}$) simplifies the last two steps. The third one becomes trivial, and the only significant matrix-vector operations occur in the last step (acceleration calculations).

The "full-step" sequence (2.24) should be avoided in practice, however, because of its poor error propagation characteristics. The so-called "half-step" or "half-station" formulation is much preferable:

$$\begin{aligned} \underline{u}_n &= \underline{h}_n^u \\ \underline{v}_{n+\frac{1}{2}} &= \dot{\underline{u}}_{n+\frac{1}{2}} = \underline{h}_{n+\frac{1}{2}}^{\dot{u}} \\ \dot{\underline{v}}_n &= \ddot{\underline{u}}_n = \underset{\sim}{M}^{-1} (\underline{Q}_n - \underset{\sim}{K}\, \underline{u}_n - \underset{\sim}{D}\, \dot{\underline{u}}_n) \end{aligned} \tag{2.25}$$

The summed form [7] of the central difference method can be presented in this form [14]. Difficulties arise, however, if the system is damped because the velocity term $\dot{\underline{u}}_n$ in (2.25c) is not readily available in this implementation. Reference [15] discusses various ways of circumventing this problem.

2.8 The Tangent Stiffness Approach

We close Section 2 with a brief review of the tangent stiffness approach to nonlinear structural mechanics. Assume that the computations have proceeded up to $t = t_n$. Introduce the increments to the next time station:

$$\Delta \underline{z}_n \overset{\text{def}}{=} \underline{z}_{n+1} - \underline{z}_n \tag{2.26}$$

where $\underline{z}$ stands for any variable vector such as $\underline{u}$, $\dot{\underline{u}}$, ..., etc. The effective force increment is linearized as follows:

$$\begin{aligned} \Delta \underline{Q}_n &= \Delta \underline{f}_n - \Delta \underline{C}_n - \Delta \underline{S}_n \\ &= \Delta \underline{f}_n - (\underset{\sim}{C}_u + \underset{\sim}{S}_u)\, \Delta \underline{u} - \underset{\sim}{C}_{\dot{u}}\, \Delta \dot{\underline{u}} + 0(||\Delta u||^2) \end{aligned} \tag{2.27}$$

in which $\underset{\sim}{C}_u$, $\underset{\sim}{C}_{\dot{u}}$ and $\underset{\sim}{S}_u$ denote the matrix Jacobians of $\underline{C}$ and $\underline{S}$ with respect to $\underline{u}$ and $\dot{\underline{u}}$, evaluated at t_n. The tangent stiffness equations can be derived as follows. Insert a Δ in front of every variable state vector in (2.3) and (2.12), replace $\Delta \underline{Q}_n$ in (2.3) by the linear portion of (2.27), eliminate $\Delta \underline{u}_n$, $\Delta \underline{v}_n$ and $\Delta \dot{\underline{v}}_n$ as in Section 2.4, and finally divide through by $h_\beta h_\delta$. The result is the "quasi-static" incremental counterpart of (2.15) (see also [3]):

$$\underset{\sim}{K}^* \Delta \underline{u}_n = \underline{H}_n + \Delta \underline{f}_n \tag{2.28}$$

where

$$\begin{aligned} \underset{\sim}{K}^* &= \underset{\sim}{K} + \underset{\sim}{C}_u + \underset{\sim}{S}_u + h_\beta^{-1} (\underset{\sim}{D} + \underset{\sim}{C}_{\dot{u}}) + (h_\delta^{-1} - h_\beta^{-1})\, \underset{\sim}{G} + (h_\beta h_\delta)^{-1}\, \underset{\sim}{M} \\ \underline{H}_n &= \left[(h_\beta h_\delta)^{-1}\, \underset{\sim}{M} + h_\beta^{-1} (\underset{\sim}{D} - \underset{\sim}{G}) \right] \hat{\underline{h}}_n^u + h_\delta^{-1}\, \underset{\sim}{A}^{-1}\, \hat{\underline{h}}_n^v \\ \hat{\underline{h}}_n^u &= \underline{h}_{n+1}^u - \underline{u}_n + h_\beta\, \dot{\underline{u}}_n \\ \hat{\underline{h}}_n^v &= \underline{h}_{n+1}^v - \underline{v}_n + h_\delta\, \dot{\underline{v}}_n \end{aligned} \tag{2.29}$$

In (2.28), $\underset{\sim}{K}^*$ is the tangent (dynamical) stiffness and $\underline{H}_n$ is a historical force vector.

In the "purely incremental" application of this approach, Eq. (2.28) is applied at each step to advance the solution by $\Delta \underline{u}_n$ and no correction is applied at t_{n+1} to account for the truncation error in (2.27). In large two- and three-dimensional problems, this naive procedure incurs in enormous computational expense in forming and factoring $\underset{\sim}{K}^*$ at each step while achieving only first-order accuracy (regardless of the accuracy of the integration operator). A better strategy is to apply one or more corrections at t_{n+1}. This helps in balancing the accuracy of the integration formula with that of the nonlinear force calculations.

If $\underset{\sim}{K}^*$ is maintained constant over many steps and recomputed only as needed because of strong nonlinearities, a restricted, "tangent-updating" version of the pseudo-force method results. As noted in Section 1.6, a more general strategy is emphasized in this paper. By judicious selection of the reference "secant" matrices in (1.1), highly nonlinear problems have been processed without a single recalculation of the left-hand side matrix $\underset{\sim}{E}$ in (2.15).

Section 3

INTEGRATION METHODS

3.1 General

Three major aspects of integration methods for nonlinear structural problems are covered in this section: stability and accuracy characteristics of integration formulas, stepsize changing techniques, and automatic stepsize selection strategies. Throughout this section we use the term "method" to represent the synthesis of these three aspects.

The question of stability of implicit formulas for nonlinear problems is far from settled and will undoubtedly continue to receive much attention. A review of some promising results is given in Section 3.2

The bulk of the literature on integration methods is concerned only with algorithmic properties (stability and accuracy) of fixed-step integration formulas. In practice, these properties are seldom preserved when the stepsize varies. Moreover, in implicit methods, the dynamic coefficient matrix has to be formed and factored whenever the stepsize is changed. We use here the term "step changing technique" to denote the set of rules and procedures for minimizing the effect of stepsize changes on stability and accuracy of the fixed-step formula, and for reducing the computational effort involved in each stepsize change. This is dealt with in Section 3.3.

Automatic stepsize selection strategies are the most talked about but least advanced of the three aspects. Two crucial questions in computational structural dynamics are: How can an explicit solution process be kept stable when low accuracy is requested, and how can an apparently stable solution computed by implicit methods be guaranteed accuracy? In Section 3.4, we present a promising step selection strategy for explicit methods which has been shown to be both robust and reliable, and which is aimed to answer the first question.

For implicit methods we also list promising error control criteria and give an appraisal based on somewhat limited experience.

In Section 3.5 we mention some of the new approaches aimed to solving effectively special classes of problems. Finally, we list some of the promising areas for future research in Section 3.6.

3.2 Stability and Accuracy of Multistep Methods

Let us consider the homogeneous, undamped nonlinear case with identity mass matrix

$$\ddot{\underline{u}} + \underline{Q}(\underline{u}) = 0 \tag{3.1}$$

Following Prothero and Robinson [16], we approximate (3.1) in the neighborhood of the numerical solution $\underline{u} = \underline{y}$ as

$$\ddot{\underline{u}} \approx -\underline{Q}(y) - (\underset{\sim}{\omega}^N)^2 (\underline{u} - \underline{y}) \tag{3.2}$$

$$(\underset{\sim}{\omega}^N)^2 = \frac{\partial}{\partial u} \underline{Q}(u) \tag{3.3}$$

The (C1) implementation of (3.2) using (2.16), i.e., the same integration formula for $\underline{u}$ and $\dot{\underline{u}}$, yields after some manipulation

$$\sum_{i=0}^{k} \underset{\sim}{A}_i \underline{\epsilon}_{n-i} = \underline{T}_n$$

where

$$\underset{\sim}{A}_i = \begin{bmatrix} \alpha_i \underset{\sim}{I} & -\beta_i h \underset{\sim}{I} \\ \underset{\sim}{\omega}^2_{n-i} \beta_i h & \alpha_i \underset{\sim}{I} \end{bmatrix}$$

$$\underline{\varepsilon}_{n-i} = \begin{Bmatrix} (\underline{u}_{n-i} - \underline{y}_{n-i}) \\ (\dot{\underline{u}}_{n-i} - \dot{\underline{y}}_{n-i}) \end{Bmatrix} \tag{3.4}$$

$$\underline{T}_n = \begin{Bmatrix} \sum_{j=0}^{k} (\alpha_i \underline{y}_{n-i} - h\beta_i \dot{\underline{y}}_{n-i}) \\ \sum_{j=0}^{k} (\alpha_i \dot{\underline{y}}_{n-i} + h\beta_i \underline{Q}_{n-i}) \end{Bmatrix}$$

The stability of multistep formulas for model nonlinear equations (3.1) can be assessed from the characteristic equation

$$\det \left| \sum_{i=0}^{k} \lambda^{k-i} \underline{A}_i \right| = 0 \quad ; \quad |\lambda_i| \leq 1 \tag{3.5}$$

and the local error is obtained from (3.4) by taking $\{\varepsilon_{n-j} = 0;\ j=1,..,\ k\}$.

$$\underline{\ell}_c = \underline{A}_o^{-1} \underline{T}_n \tag{3.6}$$

We now deduce some known results from (3.4 - 3.6) when Equation (3.1) reduces to a scalar problem. For the trapezoidal rule ($\alpha_o = -\alpha_1 = 1$, $\beta_o = \beta_1 = 0.5$), Equation (3.5) becomes

$$(\lambda - 1)^2 + \frac{h^2}{4} (\lambda+1) \left(\omega_n^2 \lambda + \omega_{n-1}^2\right) = 0 \tag{3.7}$$

from which one obtains the local stability condition of the trapezoidal rule

$$\omega_n \geq \omega_{n-1} \tag{3.9}$$

The stability condition (3.8) implies that the trapezoidal rule is locally unstable for spring-softening in the sense that the solution perturbation grows (see Fig. 5), viz.,

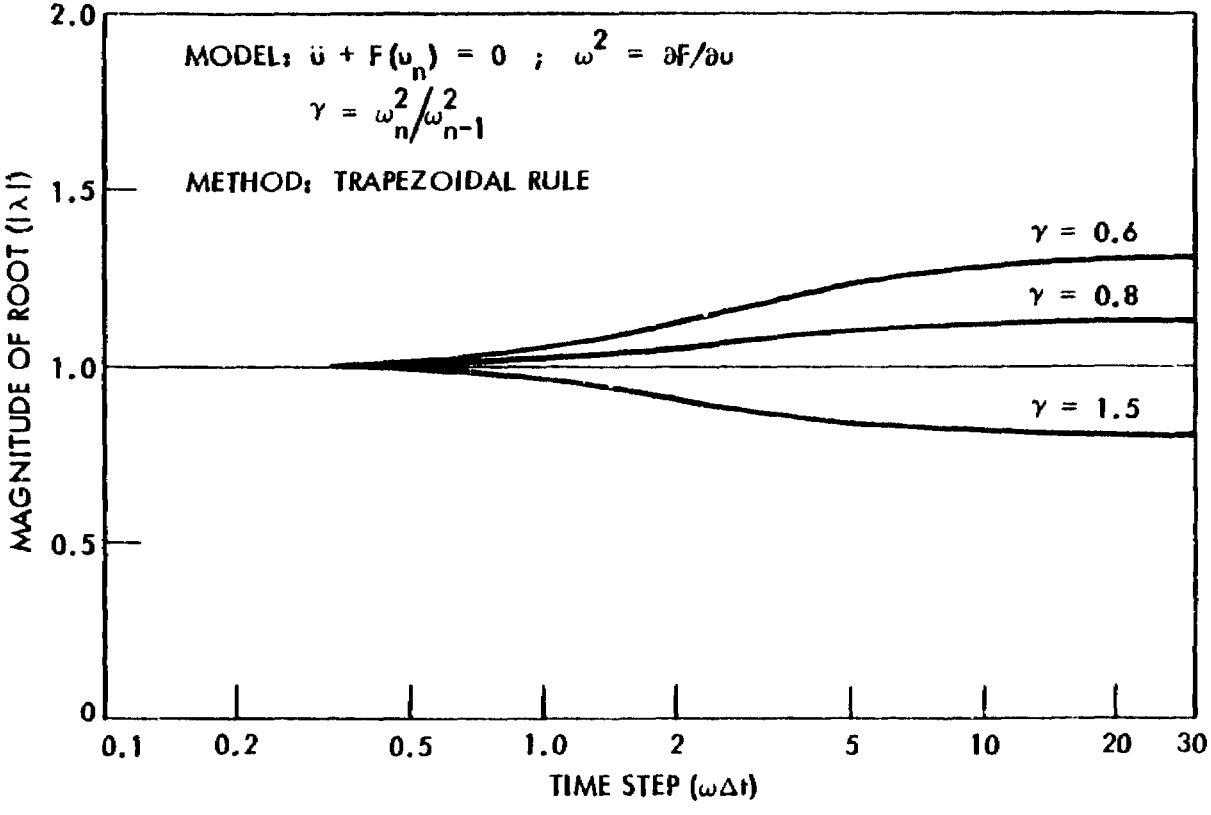


Fig. 5 Local Error Amplification Factor of Trapezoidal Rule for a Model Nonlinear Problem

$$\left| \frac{u_{n-1} - y_{n-1}}{u_n - y_n} \right| > 1 \tag{3.9}$$

This undesirable property was pointed out by Gourlay [17] for first-order parabolic equations and for second-order hyperbolic equations by Park [18,19], in which a pathological case was presented. For bilinear-spring models, several pathological examples were presented by Hughes [20]; detailed conditions for unstable solutions were recently derived by Westermo [21] by the phase-plane technique.

The local error is easily computed from (3.16)

$$\begin{aligned} \ell_c &= \left(1 + \frac{h^2\omega_n^2}{4}\right)^{-1} \left\{ y_n - y_{n-1} - h\,\dot{y}_{n-1} + \frac{h^2}{4}\,(Q_n+Q_{n-1}) \right\} \\ &= \frac{h^2}{4} \left(1 + \frac{2\omega_n^2}{4}\right)^{-1} \left\{ (Q_n-Q_{n-1}) + \frac{2}{3}\,h\,\ddot{y}_{n-1} \right\} \end{aligned} \tag{3.10}$$

from which two asymptotic cases are obtained

$$\ell_c \approx \begin{cases} \dfrac{h^2}{4}\,(Q_n-Q_{n-1}) + \dfrac{h^3}{6}\,\ddot{y}_{n-1} \quad , \quad \omega_n\, h \to 0 \\[2ex] \dfrac{h^2}{(\omega_n h)^2} \left[(Q_n-Q_{n-1}) + \dfrac{2}{3}\,h\,\ddot{y}_{n-1} \right] \quad , \quad \omega_n\, h \to \infty \end{cases} \tag{3.11}$$

The asymptotic expressions (3.11) show that the local error can be made small for the two extreme frequency components. However, nothing can be said from (3.10) as regards the magnitude of the local error corresponding to intermediate frequency components. In a practical context, this implies that even though the stability condition (3.8) is violated for the two extreme frequency components, the absolute error magnitude will most likely remain small and may not be detected. This suggests that the instability of the numerical solution by the trapezoidal rule becomes noticeable when it is caused by intermediate frequency components as explained in [22].

The source of the instability of the trapezoidal rule for certain nonlinear problems was attributed to the use of historical derivatives in [18]. The effect of historical derivatives can be eliminated by adopting <u>one-leg formulas</u>, which are obtained by selecting

$$\bar{\beta}_o = \frac{1}{k} \sum_{j=0}^{k} \beta_j \quad , \quad \bar{\beta}_j = 0; \quad j = 1, \ldots, k \tag{3.11}$$

Notice that backward difference formulas are a special class of the one-leg formulas. Let us consider the midpoint implicit formula, which is the one-leg adaptation of the trapezoidal rule, viz.,

$$\begin{aligned} \hat{u}_n &= u_{n-1} + \frac{h}{2} \dot{\hat{u}}_n \\ \dot{\hat{u}}_n &= \dot{u}_{n-1} + \frac{h}{2} \ddot{\hat{u}}_n \end{aligned} \tag{3.12}$$

where $\hat{u}_n = \frac{1}{2} (u_n + u_{n-1})$

The characteristic equation for the midpoint rule (3.12) is

$$(\lambda-1)^2 + \frac{\omega_n^2 h^2}{4} (\lambda+1)^2 = 0 \tag{3.13}$$

which indicates stability regardless of the possible variation of ω_n. The local error is obtained in the form

$$\begin{aligned} \ell_c &= \left(1 + \frac{\omega_n^2 h^2}{4}\right)^{-1} \left\{ y_n - y_{n-1} + h \dot{y}_{n-1} - \frac{h^2}{2} Q(\hat{y}_n) \right\} \\ &= \frac{h^2}{2} \left(1 + \frac{\omega_n^2 h^2}{4}\right)^{-1} \left\{ \ddot{y}_{n-1} + Q(\hat{y}_n) + \frac{h}{3} \dddot{y}_{n-1} \right\} \\ &= -\frac{h^2}{4} \left(1 + \frac{\omega_n^2 h^2}{4}\right)^{-1} \ddot{y}_n \end{aligned} \tag{3.14}$$

Therefore,

$$\ell_c \approx \begin{cases} -\frac{h^2}{4}\ddot{y}_n & , \quad \omega_n h \to 0 \\ \frac{h^2 \ddot{y}_n}{(\omega_n h)^2} & , \quad \omega_n h \to \infty \end{cases} \tag{3.15}$$

The comparison of the two asymptotic error expressions (3.11) and (3.15) shows that the trapezoidal rule is more accurate than the midpoint rule for nonlinear problems provided

$$|\varrho| \sim o(|\ddot{y}|) \tag{3.16}$$

The two one-step formulas have been used to integrate the model nonlinear problem (see Park [18])

$$\ddot{u} + 100 \tanh(u) = 0, \quad u(0) = 0, \quad \dot{u}(0) = 25 \tag{3.17}$$

The results show that the midpoint rule gives stable solutions for all stepsize ranges tested while the solutions by the trapezoidal rule becomes unstable for stepsizes $h > 0.2$. The interested reader is referred to recent work by Dahlquist [23] for the stability analysis of multistep one-leg formulas.

3.3 Step Changing Technique

A proper choice of implicit integration formulas can ensure numerical stability without stepsize limitations (the so-called A-stability). The stepsize is then established solely from accuracy considerations. Exploitation of this favorable property is often difficult, however, because of the computational overhead associated with stepsize changes. These difficulties are examined in Section 3.3.1, which is followed by the presentation of an improved matrix scaling technique [24]. This technique accommodates stepsize changes without requiring a recalculation of the matrix $\underline{E}$ in (2.15). Applicable ranges of this technique and the importance of associated predictors are examined in Section 3.3.3. Finally, the effect of stepsize changes on the stability of the integration formula is discussed in Section 3.3.4.

3.3.1 Difficulties in Variable-Step Implicit Methods

We recall Equations (2.15) and (2.18) for the reader's convenience

$$\begin{aligned} \underset{\sim}{E}\, \underline{u}_n &= \underline{g}_n \\ \underset{\sim}{E}(\delta) &= \underset{\sim}{M} + \delta\, \underset{\sim}{D} + \delta^2\, \underset{\sim}{K} \quad , \quad \delta = h_3 = h \\ \underline{g}_n &= \underset{\sim}{M}\, \underline{h}_n^u + \delta\, \underline{h}_n^v + \delta^2\, \underline{Q}_n \end{aligned} \tag{3.17}$$

The stability properties of the integration formula (2.17) can be maintained by solving (3.17a) accurately by a convergent iterative method. If the mass matrix $\underset{\sim}{M}$ is diagonal, the following predictor-corrector iteration scheme appears at first sight attractive:

$$\underline{u}_n^{(m+1)} = \underset{\sim}{M}^{-1} \left[\underline{g}_n^{(m)} - \delta(\underset{\sim}{D} + \delta\, \underset{\sim}{K})\, \underline{\dot{u}}_n^{(m)} \right] \tag{3.18}$$

(This results from splitting $\underset{\sim}{E} = \underset{\sim}{M} + (\delta\, \underset{\sim}{D} + \delta^2\, \underset{\sim}{K})$ and transferring the second term to the right-hand side.) Unfortunately, it is well known that (3.18) converges only if

$$c\, \omega_{max}\, h < 1 \tag{3.19}$$

where c is a constant of order unity. The stepsize restriction (3.19) is of the same order as that imposed by explicit methods. Thus, (3.18) wipes out the basic advantage of implicit integration. The A-stability of implicit formulas can be retained by using the Newton-like iteration

$$\underset{\sim}{E}^{(m)}(\delta) \left(\underline{u}_n^{(m+1)} - \underline{u}_n^{(m)} \right) = \underline{g}_n^{(m)} - \underset{\sim}{E}^{(m)}(\delta)\, \underline{u}_n^{(m)} \tag{3.20}$$

Notice that for linear problems, i.e., constant $\underset{\sim}{E}$ and $\underline{g}_n$ (3.20) gives the converged solution with one iteration regardless of the predictor $\underline{u}_n^{(0)}$. If the stepsize is varied, the matrix $\underset{\sim}{E}(\delta_{new})$ has to be refactored. This can be costly in practice and is a major barrier to the effective use of implicit formulas for variable step integration of the equations of motion. A means to overcome this difficulty is described below.

3.3.2 Matrix Scaling Technique

Let us express the matrix $\underset{\sim}{E}(\delta)$ in the form

$$\underset{\sim}{E}(\delta) = \frac{1}{s}\ \underset{\sim}{E}(\delta_o) - \underset{\sim}{R} \tag{3.21}$$

where s is a scaling parameter and $\underset{\sim}{R}$ is a residual matrix to be determined. Substitution of (3.21) into (3.17a) yields

$$\underset{\sim}{E}(\delta_o)\ \underline{u}_n - s\ (\underline{q}_n + \underset{\sim}{R}\ \underline{u}_n) = 0 \tag{3.22}$$

where the residual matrix $\underset{\sim}{R}$ is

$$\underset{\sim}{R} = \left(\frac{1}{s} - 1\right)\underset{\sim}{M} + \left(\frac{\delta_o}{s} - \delta\right)\ \underset{\sim}{D} + \left(\frac{\delta_o^2}{s} - \delta^2\right)\underset{\sim}{K} \tag{3.23}$$

An iterative method for (3.22) takes the form

$$\underline{u}_n^{(m+1)} = s\ \underset{\sim}{E}\ (\delta_o)^{-1}\ \ (\underline{q}_n + \underset{\sim}{R}\ \underline{u}_n^{(m)}) \tag{3.24}$$

Notice that we have freedom for improving this iterative method by choosing the scaling parameter s and predictor $\underline{u}^{(m)}$ appropriately. The convergence of the iterative method (3.24) can be conveniently examined using the scalar case

$$\underset{\sim}{M} = 1, \quad \underset{\sim}{D} = 2\xi\omega \quad \text{and} \quad \underset{\sim}{K} = \omega^2 \tag{3.25}$$

The spectral radius of the iteration matrix becomes

$$\kappa\left[s\ E^{-1}(\delta_o)\ R\right] = \frac{\left|(1-s) + 2(\delta_o - s\delta)\ \xi\omega + (\delta_o^2 - s\delta^2)\ \omega^2\right|}{1 + 2\ \xi\omega\delta_o + \omega^2\delta_o^2} \tag{3.26}$$

a) Fixed-step case ($s=1,\ \delta_o = \delta$). The spectral radius is zero from (3.26) and accordingly the solution converges with one iteration as it should.

b) Variable Step Case ($\delta_o \neq \delta$). This corresponds to integrating with a stepsize other than the factored stepsize δ_o. If the system is undamped ($\xi = 0$), the optimal scaling parameter is

$$s = \frac{1 + \delta_o^2 \omega^2}{1 + \delta^2 \omega^2} \tag{3.27}$$

For the multidimensional case, the choice (3.27) with $\omega = \omega_c$, a "scaling frequency", gives

$$\kappa \simeq \frac{|(\eta^2 - 1)(\Omega - \Omega_c^2)|}{(1 + \Omega_c^2)(1 + \eta^2\Omega^2)} \tag{3.28}$$

in which $\eta = \delta_o/\delta$, $\Omega = \delta\omega$.

The case $\Omega_c = 0$ and $\eta \geq 1$ was investigated in [25] and some preliminary performance data was reported in [5]. If $\eta < 1$, (3.28) shows that $\Omega_c \sim \Omega_{max}$ is probably adequate.

Global convergence for the entire frequency range is obtained with the following choice of the scaling parameter:

$$\kappa = \begin{cases} \dfrac{(\eta^2 - 1)\,\Omega^2}{1 + \eta^2\,\Omega^2} & , \quad s = 1 \ , \ \eta \geq 1 \\[2ex] \dfrac{(1 - \eta^2)(\Omega_{max}^2 - \Omega^2)}{(1 + \Omega_{max}^2)(1 + \eta^2\Omega^2)} & , \quad s = \dfrac{1 + \eta^2\Omega_{max}^2}{1 + \Omega_{max}^2} \ , \ \eta < 1 \end{cases} \tag{3.29}$$

As can be seen from (3.29a), the spectral radius κ for $\eta \geq 1$ rapidly approaches unity as the sampling frequency Ω increases. This restricts the choice of predictors to extrapolation from displacements only, because the inclusion of velocity terms in the predictor can generate large errors in the predicted displacements which contain high-frequency components.

On the other hand, when $\eta = \delta_o/\delta < 1$, the spectral radius of (3.29b) approaches unity and zero for low and high frequency components, respectively. A desirable convergence property is to obtain as small a spectral radius as possible for the "accuracy" range $0 \leq \omega h \leq 2$ and to bound the spectral radius below

unity for the "noise" range $\omega h > 2$ for iteration stability. Note, for example, that the globally convergent choices (3.29) yield

$$\kappa = \begin{cases} \dfrac{(\eta^2 - 1)}{1 + \eta^2} \to 1 & , \quad \eta >> 1 \; ; \quad \Omega \to 1 \\ \dfrac{(1 - \eta^2)\ (\Omega_{max}^2 - 1)}{(1 + \Omega_{max}^2)\ (1 + \Omega^2)} \to 1 & , \quad \eta << 1 \; ; \quad \Omega \to 1 \end{cases} \tag{3.30}$$

This indicates that both cases give a poor convergence rate in the accuracy range as the integrating stepsize δ deviates further from the factored stepsize δ_o.

3.3.3 Improvement of Low-Frequency Convergence Rate

In order to improve the convergence rate for the accuracy range $0 \leq \omega h \leq 1$, we introduce an acceleration scheme

$$\underline{u}^{(m+1)} = \alpha\, \underline{u}^{(m)} + (1 - \alpha)\, \underline{u}^{(m-1)} - \beta\, \underline{r}^{(m+1)} \tag{3.31}$$

where

$$\underline{r}^{(m+1)} = \underline{g} - s\, \underline{\underline{E}}\, \underline{\underline{E}}_o^{-1} (\underline{g} + \underline{\underline{R}}(s)\, \underline{u}^{(m)}) \tag{3.32}$$

Combining (3.31) with (3.24) one can derive the following error equation

$$\underline{r}^{(m+2)} = \alpha\, \underline{r}^{(m+1)} + (1 - \alpha)\, \underline{r}^{(m)} - \beta\, s\, \underline{\underline{E}}\, \underline{\underline{E}}_o^{-1}\, \underline{\underline{R}}\, \underline{r}^{(m+1)} \tag{3.33}$$

The stability and convergence properties of the acceleration scheme (3.31) can be evaluated from its associated characteristic equation

$$\lambda^2\, \underline{\underline{I}} - (\alpha\, \underline{\underline{I}} - \beta\, s\, \underline{\underline{E}}\, \underline{\underline{E}}_o^{-1}\, \underline{\underline{R}})\, \lambda + (\alpha - 1)\, \underline{\underline{I}} = 0 \tag{3.34}$$

It can be shown [24] that stability requires that

$$0 < \alpha < 2,\ \beta\ \frac{(1+\Omega^2)\left[(1-s)+(\eta^2-s)\,\Omega^2\right]}{(1+\eta^2\,\Omega^2)} \leq 2\alpha \tag{3.35}$$

The conditions (3.35) are satisfied by the choice

$$\alpha = \frac{\mu_{max}+\mu_{min}}{2}\beta\ ,\ \beta = \frac{4}{\sqrt{\mu_{min}}+\sqrt{\mu_{max}}} \tag{3.36}$$

where μ's are the eigenvalues of the system

$$(s\ \underline{E}(\delta)\ \underline{E}(\delta_o)^{-1}\ \underline{R} - \mu\ \underline{I})\ \underline{Y} = \underline{0} \tag{3.37}$$

The convergence rate κ of the acceleration scheme (3.31) can be expressed as

$$\kappa = \frac{1-\sqrt{\mu_{max}/\mu_{min}}}{1+\sqrt{\mu_{max}/\mu_{min}}} \tag{3.38}$$

Consequently, the narrower the spread of the eigenvalue spectrum, the faster the convergence rate.

When the integration stepsize δ is larger than the factored stepsize δ_o, i.e., $\eta = \delta_o/\delta < 1$, one obtains

$$\mu = \begin{cases} \dfrac{(1-\eta^2)\,a^2}{1+a^2} \to 1 & ,\quad \Omega \to 0 \\ \dfrac{(1-\eta^2)\,\rho}{1+\epsilon\eta^2} \to \rho & ,\quad \Omega \to \infty \\ 0.17(1-\eta^2)/\eta^2 & ,\quad \Omega_{cr}^2 \to 0.44/\eta^2\ ;\ \Omega_{max}\,\eta = 1 \end{cases} \tag{3.39}$$

where we have chosen the scaling parameter s and the constant a^2 as

$$s = \frac{1+\eta^2 a^2}{1+a^2} \tag{3.40}$$

$$a^2 = (1 + \rho\ \eta^2)\ \Omega_{max}^2 \tag{3.41}$$

The preceding asymptotic estimates of the eigenvalues of (3.37) indicate that if Ω_{max} is accurately known and $\Omega_{max}\ \delta_o/\delta \approx 1$, one can achieve a rapidly convergent scheme through the acceleration (3.31).

A similarly favorable convergence rate for the case $\eta > 1$ is harder to achieve and requires alternating iterations with two different factored matrices [24].

3.34 Effect of Step Changes on Stability

There are two techniques for incorporating step changes in the multistep

integration formulas: interpolation of past solutions to represent solutions at equal intervals of the present stepsize and use of variable coefficients in the integration formulas. In general, interpolation requires additional solution vectors to be stored for preservation of the fixed-step accuracy and stability. The introduction of variable coefficients alleviates the additional storage requirement, but is known to cause instability if the stepsize is varied too often [26]. Figure 6 shows the stability region of a three-step formula [18] as adopted to variable coefficient technique in the form

$$\alpha_o \, u_{n+1} = \alpha_1 \, u_n + \alpha_2 \, u_{n-1} + \alpha_3 \, u_{n-2} + h \, \dot{u}_{n+1}$$

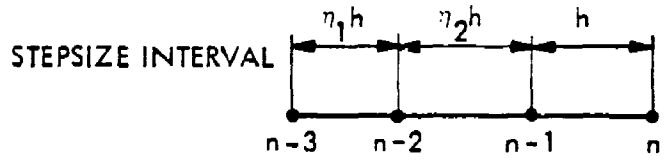
where

$$\begin{aligned}
\alpha_o &= \frac{(2 + \eta_1)}{1 + \eta_1} + \frac{1}{2} \frac{1}{(1 + \eta_1 + \eta_2)} \\
\alpha_1 &= \frac{(1 + \eta_1)}{2\eta_1} \frac{(1 + 2\eta_1 + 2\eta_2)}{(\eta_1 + \eta_2)} \\
\alpha_2 &= - \frac{1}{2\eta_1^2} \frac{(1 + 3\eta_1)}{(1 + \eta_1)} \\
\alpha_3 &= \frac{1}{2} \frac{(1 + \eta_1)}{(1 + \eta_1 + \eta_2)(\eta_1 + \eta_2)\eta_2}
\end{aligned} \tag{3.42}$$

and

$$\eta_1 = h_n / h_{n+1} \quad , \quad \eta_2 = h_{n-1} / h_{n+1} \, .$$

Note from Figure 6 that a decrease in the stepsize shrinks the unstable zone and results in an increased numerical damping as the stability boundary departs further from the imaginary axis. An increase in the stepsize, on the other hand, expands the unstable zone which includes a segment of the imaginary axis. This segment will cause local instability if the combination of frequency contents and the stepsize,



	A	B	C	D	E	F
η_1	1.0	2.0	1.0	0.5	1.0	0.625
η_2	1.0	2.0	2.0	0.5	0.5	0.625

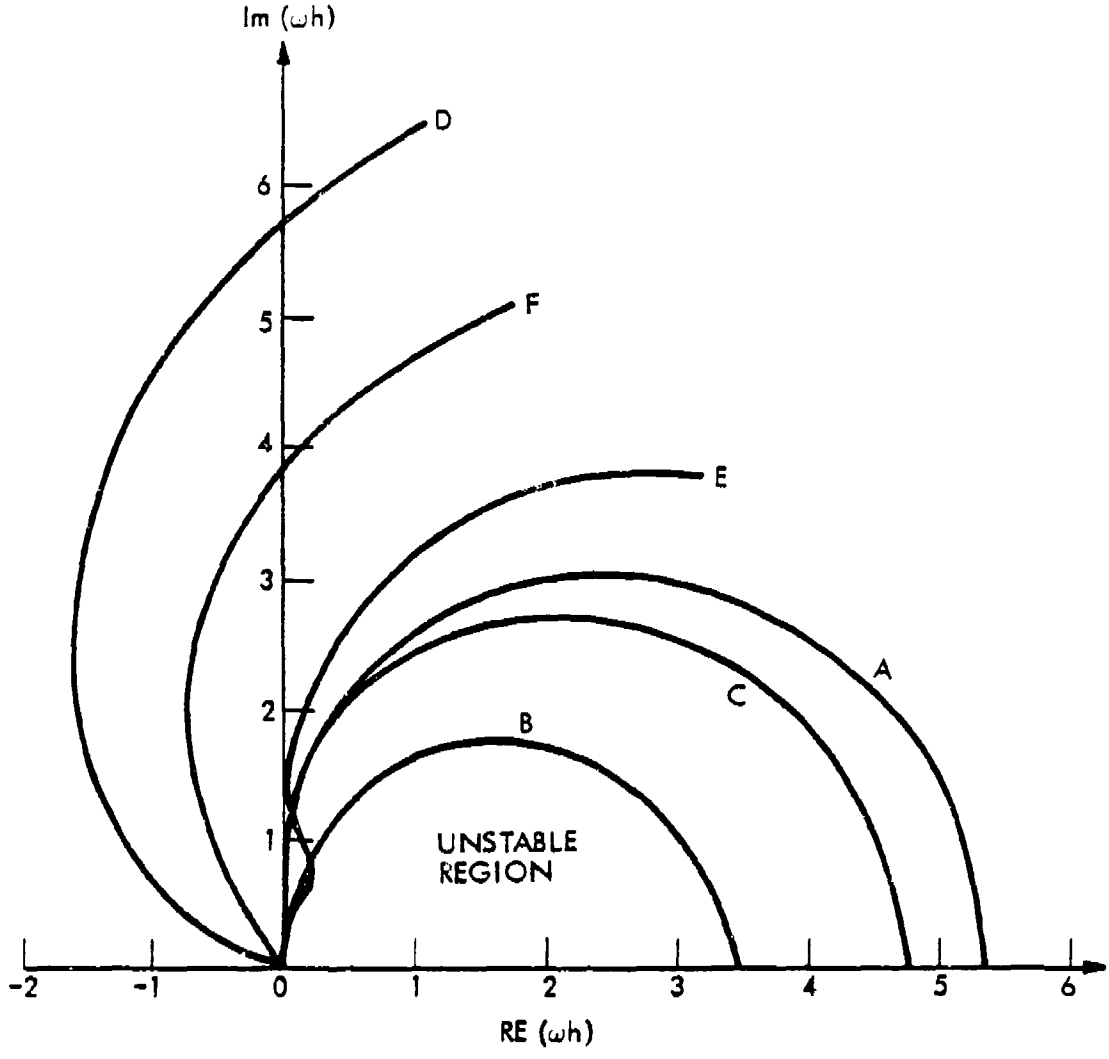


Fig. 6 Effect of Uneven Step Interval on Local Stability

ωh, falls in that zone. It is also noted that the fixed, k-step formula is recovered at k-th step after the stepsize is changed.

How seriously stepsize changes affect the global stability and accuracy has not been systematically studied yet. Experience indicates that, for k-step formulas, the effect is insignificant if the stepsize is varied at most every 2k-th step.

3.3.5 Step Changing Techniques for the Central Difference Formula

The effect of step changing techniques on the stability of explicit formulas is more pronounced than on implicit formulas because the stability margin of explicit formulas is quite narrow, mostly $\omega h \leq 2.0$. We present a relevant portion of the results from [27] for the central difference formula. Of the four techniques considered (see Table 5), the velocity averaging technique (FAVE) is most desirable for undamped problems, whereas the variable coefficient technique (FVCO) outperforms the rest for heavily damped problems. As in the case of implicit formulas, the question of how often the stepsize can be changed without a noticeable affect on the global solution accuracy has not been satisfactorily resolved yet.

3.4 Stepsize Selection Strategies

The earliest known numerical integration method is due to Euler [28,29]. By 1973, the number of publications had grown to over 600 [30] and over 50 articles are published every year. Of these, at least a dozen are directly concerned with the class of stiff-oscillatory systems to which structural dynamic problems belong. Adequate stepsize selection strategies for non-stiff parabolic problems exist [31] and some promising strategies for stiff parabolic problems have been published [32]. The authors are not aware, however, of many publications that deal with stepsize selection for stiff oscillatory problems.

The paucity of publications is apparently due to the fact that low-accuracy solutions (e.g., two correct digits) are acceptable for many structural

Table 5

FORMULAS FOR STEP CHANGES

Technique	Formula
Variable Coefficient (FVCO)	$\dot{u}^{n+\frac{1}{2}} = \dot{u}^{n-\frac{1}{2}} + \frac{1}{2}(h_n + h_{n-1})\,\ddot{u}^n$ $u^{n+1} = u^n + h_n\,\dot{u}^{n+\frac{1}{2}}$
Average Velocity (FAVE)	$\dot{u}^{n+\frac{1}{2}} = \frac{1}{2}\left[(3-r)\,\dot{u}^{n-\frac{1}{2}} - (1-r)\,u^{n-\frac{3}{2}}\right] + h_n\ddot{u}^n$ $u^{n+1} = u^n + h_n\,\dot{u}^{n+\frac{1}{2}}$
Extrapolation (FEXT)	$\dot{u}^{n+\frac{1}{2}} = \frac{1}{8}\left[(7+r^2)\,\dot{u}^{n-\frac{1}{2}} - (1-r^2)\,\dot{u}^{n-\frac{3}{2}}\right] + h_n\ddot{u}^n$ $u^{n+1} = u^n + h_n\,\dot{u}^{n+\frac{1}{2}}$
Interpolation (FINT)	$\dot{u}^{n+\frac{1}{2}} = \dot{u}^{n-\frac{1}{2}} + \frac{1}{8}(h_n + h_{n-1})\left[(3+r)\ddot{u}^n + (1-r)\,\ddot{u}^{n-1}\right]$ $u^{n+1} = u^n + h_n\,\dot{u}^{n+\frac{1}{2}}$

1) $r = h_n/h_{n-1}$

2) All the above formulas recover the basic formula (2.25) when $r = 1$.

dynamic problems, while most of the available stepsize selection strategies are based on the truncation error concept, which is effective chiefly for high-accuracy requirements.

For variable-step explicit integration, the ability of detecting local (incipient) instability is mandatory. For implicit integration, the ability of maintaining a given accuracy level is essential. A general-purpose integration package should therefore have the ability to monitor both cases depending on the integration mode, and transition from one mode to the other should be governed by anticipated cost minimization. We now examine these goals in more detail.

3.4.1 Explicit Variable-Step Strategies

In low-accuracy explicit integration, the detection of instability becomes a major concern. This requires an accurate assessment of the highest frequency of the system. Experience shows that this is not a simple task in nonlinear problems, and repeated calculations to that effect can be costly. Numerical experiments have shown that monitoring the truncation error as stability guard is generally useless. Of several empirical techniques proposed and tested so far, the following "perturbed apparent local frequency" concept has been found to be reliable and efficient. The perturbed apparent frequency ω_a is defined as

$$\omega_{aj}^2 = - \frac{\Delta u_i \, \Delta \ddot{u}_j}{\Delta u_j^2} \tag{3.43}$$

which can be expressed for linear homogeneous equations as [27]

$$\omega_{aj}^2 = \frac{\sum \omega_j^2 \, t_{jm}^2 \, \Delta q_m^2}{\sum t_{jm}^2 \, \Delta q_m^2} \quad , \quad m = 1, \ldots, n_f \tag{3.44}$$

where t_j is the i-th column eigenvector and $\underline{q}$ is the uncoupled modal displacement vector from the relationship

$$\underline{u} = \underset{\sim}{T}\,\underline{q} \tag{3.45}$$

Remark 1. The qualifier "apparent" is used here to emphasize only those frequencies that are participating in the dynamic response. For example, a structure under free-fall gravity motion exhibits only the rigid-body motion (zero frequency). In this case, $(\omega_a)_{max}$ is zero and any arbitrarily large stepsize should yield an exact solution, which is confirmed by numerical experiments [33].

Remark 2. The highest frequencies participate in the dynamic response at early time and may hardly participate at a later time. The apparent frequency concept should in this case result in an increased stepsize at the late time period, which appears also to be the case.

Numerical experiments indicate that the step selection strategy based on the apparent-frequency sampling theory outperforms all other strategies tested. Further refinements of this strategy are expected by experimenting on large-scale problems.

3.4.2 Implicit Variable-Step Strategies

Four strategies for stepsize selection in implicit integration will be mentioned. They are based on the local truncation error, residual term, Richardson's extrapolation and deviation from linearity, respectively.

The use of the truncation error is well known and need not be further explained, see e.g. [7]. The residual term $\underline{d}$ is the difference between the acceleration computed from the differential equations of motion (1.1) and that computed from the integration formula (2.7b):

$$\underline{d}_n = \ddot{\underline{u}}(\dot{\underline{u}}_n, \underline{u}_n) - (\dot{\underline{u}}_n - \dot{\underline{u}}_n^{h})/h_\delta \tag{3.46}$$

An approximate local error $\underline{\varepsilon}_n$ can be expressed [34] as

$$\underline{\varepsilon}_n = \underline{u}(t_n) - \underline{u}_n \sim \frac{h^2}{p+1}\,\underline{d}_n \tag{3.47}$$

where p is the order of the integration formula.

The strategy based on Richardson's extrapolation uses the difference between two solutions obtained with stepsizes h and $\frac{1}{2}$ h, respectively, to express the local error as

$$\underline{\varepsilon}_n \sim \underline{u}_n (\tfrac{1}{2} h) - \underline{u}_n (h) \tag{3.48}$$

or a similar estimate of $\underline{\varepsilon}_n$ when smoothing is performed [33]. Limited experience suggests that the residual term estimate (3.47) is far more reliable than the truncation error. The estimate (3.48) is the most reliable of the three, but requires at least three solutions per step. In future numerical experiments, error estimation by extrapolating at every tenth step is contemplated to reduce the computational overhead.

Finally, the deviation from linearity can be used as stepsize selection strategy. This works effectively if the major source of error is due to system nonlinearities and one has a priori knowledge of the linear behavior of the system. This option is attractive in those cases where the analyst performs a linear dynamic analysis first and then decides to proceed to a nonlinear analysis on account of critical design considerations.

No consensus as to which strategy is most suitable for structural dynamic analysis has been reached as yet. In fact, this constitutes an important area of near-future research (cf. Section 3.6).

3.5 New Approaches

It should be clear from previous sections that the relative performance of explicit and implicit methods depends on problem characteristics, user requirements (especially accuracy level), and the computer cost function. The two methods can be applied concurrently over different portions of the structure [36,37] or in an interleaving manner so that the integration process alternates between the two modes according to a cost minimization criterion [38].

A combination of explicit integration and mode superposition methods can be used to increase the effective integration stepsize. Knowledge of a set of high frequency dominant eigenvectors allow the corresponding subspace to be annihilated from the equations of motion [39]. The net result of this technique is an extension of the maximum stable stepsize for explicit methods; this gain may be significant for certain classes of problems.

It may also be possible to combine the traditional Rayleigh-Ritz method with implicit methods. Direct time integration would be used only occasionally to evaluate errors in the Rayleigh-Ritz approximations and, if necessary, to trigger the injection of additional sets of trial mode shapes. Such an effort would probably shed some light on the usefulness of the so-called exponentially fitted formulas from an engineering application viewpoint.

The classes of problems for which each of the aforementioned approaches become advantageous has not been fully explored, and more extensive testing will be required before definite conclusions to this effect are reached.

3.6 Areas for Future Research

The increasing importance of dynamic analysis in design and verification of complex structures has given renewed impetus to research activities in direct time integration methods. During the past decade, significant progress has been made in the following areas: techniques for stability/accuracy assessment in linear problems, qualitative assessment of effect of nonlinearities on stability, comparative performance rating of multistep formulas on various nonlinear model problems, and spatial hybridization of explicit and implicit formulas.

Three important near-term research areas will be mentioned. The authors believe that the most important one is the further development of stepsize control strategies for implicit integration methods. Progress in this area would facilitate the production of cost effective dynamic analyzers that can operate with the same degree of reliability as that of presently available static analyzers, and hence attract interest from practicing engineers.

A second area involves the systematic study of performance crossover points between explicit and implicit methods for a given computer environment and a set of representative problems. The crossover point can be expected to depend not only upon the required accuracy level and the stepsize selection technique, but also upon the computer characteristics and the quality of the supporting software. Resolution of this topic would simplify the implementation of problem-adaptive temporal switches between both integration modes.

A third area pertains to the study of the influence of formulations and computational procedures for nonlinear problems on the stability of the integration formulas. (In other words, investigation of the interaction between Sections 2 and 3.2.)

Finally, as a long-term research topic, we should mention the potential impact of new computing equipment on the performance of direct time integration methods. Some of the techniques which are currently deemed obsolete may turn out to be most suitable, or new formulas and procedures may have yet to be developed to fit new computing environments. At the present, however, the development of new integration formulas is not likely to advance the state of the art as significantly as that of the three near-term areas.

4. CONCLUDING REMARKS

Substantial computational cost reductions, of the order of 2 to 50, have been observed as a result of the use of the generalized pseudo-force approach over the conventional tangent stiffness approach. Another cost reduction of roughly 2 to 5 can be achieved by paying careful attention to implementation details. The total gain can thus span one or two orders of magnitude.

An important consequence is that nonlinear dynamic analysis can no longer be regarded as being orders of magnitude more expensive than the corresponding linear analysis. For sufficiently large problems the array-processing effort involved in the solution-advancing cycle dominates over that of evaluating nonlinear terms. Under such circumstances, and assuming identical integration-driving software, the (nonlinear/linear) cost ratio per time step is close to unity in explicit integration. For implicit integration, the ratio depends primarily on the number of iteration cycles per step if refactorizations are avoided, and typically varies from 2 to 5. These significant cost reductions have been translated into Figure 7; this is an update of a chart prepared in 1976 [3], which reflected the then prevalent use of the tangent stiffness approach.

It is hoped that the improved computational efficiency, when coupled to the increased reliability brought about by the problem-adaptive techniques discussed in Section 3, will eventually result in broader usage of transient nonlinear analysis by practicing engineers

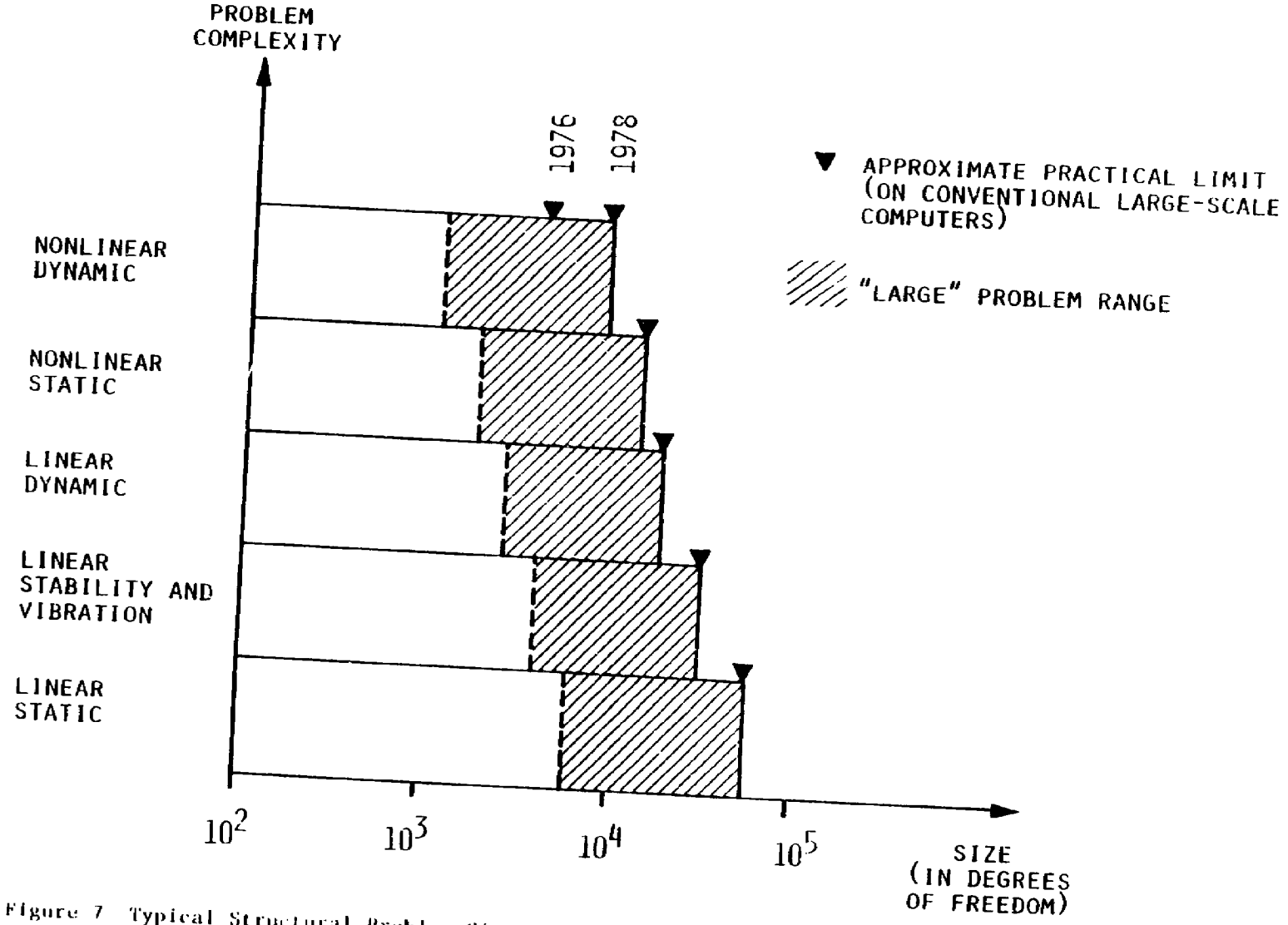


Figure 7 Typical Structural Problem Sizes that Can Be Currently Processed on Conventional Large-Scale Computers.

Appendix A

COMPUTER IMPLEMENTATION OF DYNAMIC ANALYSIS

A.1 Organizational Approaches

Two philosophies may be followed in the implementation of transient response analysis capabilities.

1. Tightly Coupled Processors. The analysis control is embedded within the framework of a specific structural analyzer.

2. Loosely Coupled Processors. The analysis control is segregated as a modular "driver" processor, which is interfaced with existing analyzers through a data base management system.

(A similar choice appears in other application programming areas such as optimization and synthesis, in which a general purpose driver processor controls analysis processors.)

The first approach offers the benefits of expedience and potentially high computational efficiency, inasmuch as special analyzer features can be exploited. It is indicated if the development is carried out by the designer of the analyzer, or a tightly knit team that includes the designer. The main disadvantage of this approach is the sacrifice in flexibility of application, maintainability (the ability to repair errors), and modifiability (the ability to make functional changes). Furthermore, if the structural analyzer is very complex, or is adjudged proprietary by its source, this approach may be unfeasible.

The second approach is consistent with present trends as regards construction of data base linked program networks [40]. Consequently, it is believed to offer the greatest potential on a long term basis. It is particularly appealing when several existing analyzers, some of which may

be unmodifiable, are to be interfaced with the same integration driver. One drawback of this approach is the need for considerable attention on the part of the developer to the overall design of the program to avoid "downstream" surprises. Separable driver operation also presupposes the ready availability of a library of supporting matrix processors as well as that of a scientific data management system. Because of the long-term nature of the development effort, this approach, which is the one discussed in the following sections, is not necessarily suitable for every programming group.

A.2 The Players

Many industrial and engineering organizations maintain a library of structural analyzers. These are either acquired from external sources or, if the organization is sufficiently large, developed in house. Most of the analyzers will be linear, some nonlinear; in the latter case, a restricted problem application range is the norm rather than the exception.

Another intervening software element is the time integration control module, herein called the integrator (I) for brevity.

The function of the analyzer (A) is to generate the discrete dynamical equations, i.e., the components of (1.1). The integrator advances the state solution by following one of the computational sequences discussed in Section 2. Computed responses may be fed back to the analyzer to regenerate nonlinear terms or to compute derived quantities such as stresses. Upon analysis completion, the computed responses are sent to a display post-processor (D). This is usually a separate module, although portions of D may be actually duplicated in the integrator to produce "display snapshots" during the analysis process.

The process just described is schematized in Figure A-1. This software organization may be regarded as an instance of modularization by program function [41].

A.3 Transient Linear Analysis

A large percentage of dynamic analysis concern linear problems. Moreover,

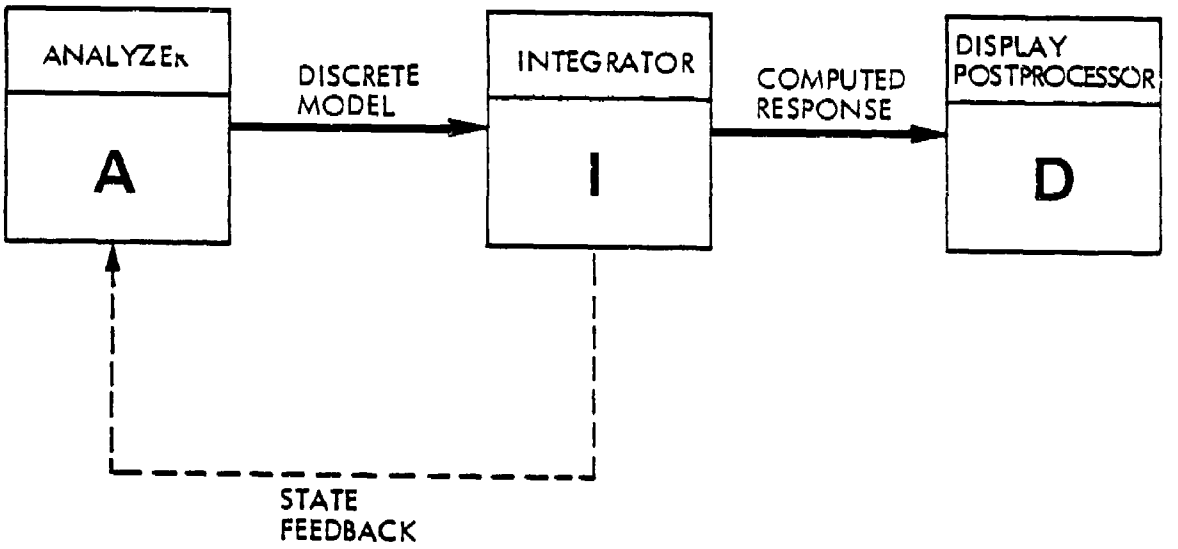


Fig. A-1 Basic Software Elements in Structural Dynamic Analysis

an important class of nonlinear problems can be processed in "piecewise linear" mode, in which the response is linear except at isolated "busy" periods. Consequently, the development of a modular, general-purpose integrator would be perhaps economically unjustifiable if a capability for efficiently functioning in linear or piecewise-linear mode is not provided. This capability is activated by suppressing nonlinear feedback.

Figure A-2 schematizes the operational flow for transient linear analysis. In this case, a high degree of modularity is possible. The structural analyzer functions primarily as an array generator. These arrays typically include the matrices $\underset{\sim}{K}$, $\underset{\sim}{D}$, and $\underset{\sim}{M}$ and the "base load vectors" $\underline{L}_i$ discussed below. These arrays are placed on a global data base (GDB), which resides on permanent storage devices. If the formatting requirements imposed by the support processing utilities (F) conflict with those of the analyzer, the arrays must be passed through a data format converter (C_1) before they are ready to be submitted to the integrator.

The temporal variation of the applied force vector can be often expressed in the separable form

$$\underline{f}(t) = \sum_{i=1}^{n_\lambda} \underline{L}_i \lambda_i(t) = \underset{\sim}{L}\,\underline{\lambda} \qquad (A.1)$$

where λ_i are user supplied load factors and $\underset{\sim}{L}$ is the base load matrix whose columns are the base load vectors $\underline{L}_i$. Frequently, $n_\lambda << n_f$, the number of freedoms; if $n_\lambda = 1$, the loading is called proportional. Usually, the base load vectors result from discretization of forcing fields such as surface tractions, temperature variations or body forces, and must be therefore generated by the structural analyzer (this is the cause of the path accessing the applied force segment in Figure A-2). The decomposition (A.1) entails considerable computational advantages if $\underset{\sim}{L}$ is constant over a set of problems and the user wishes to investigate the response to a ensemble of load histories.

Computed responses are placed on the global data base and can be later accessed by the display postprocessor module (D). If the history of derived

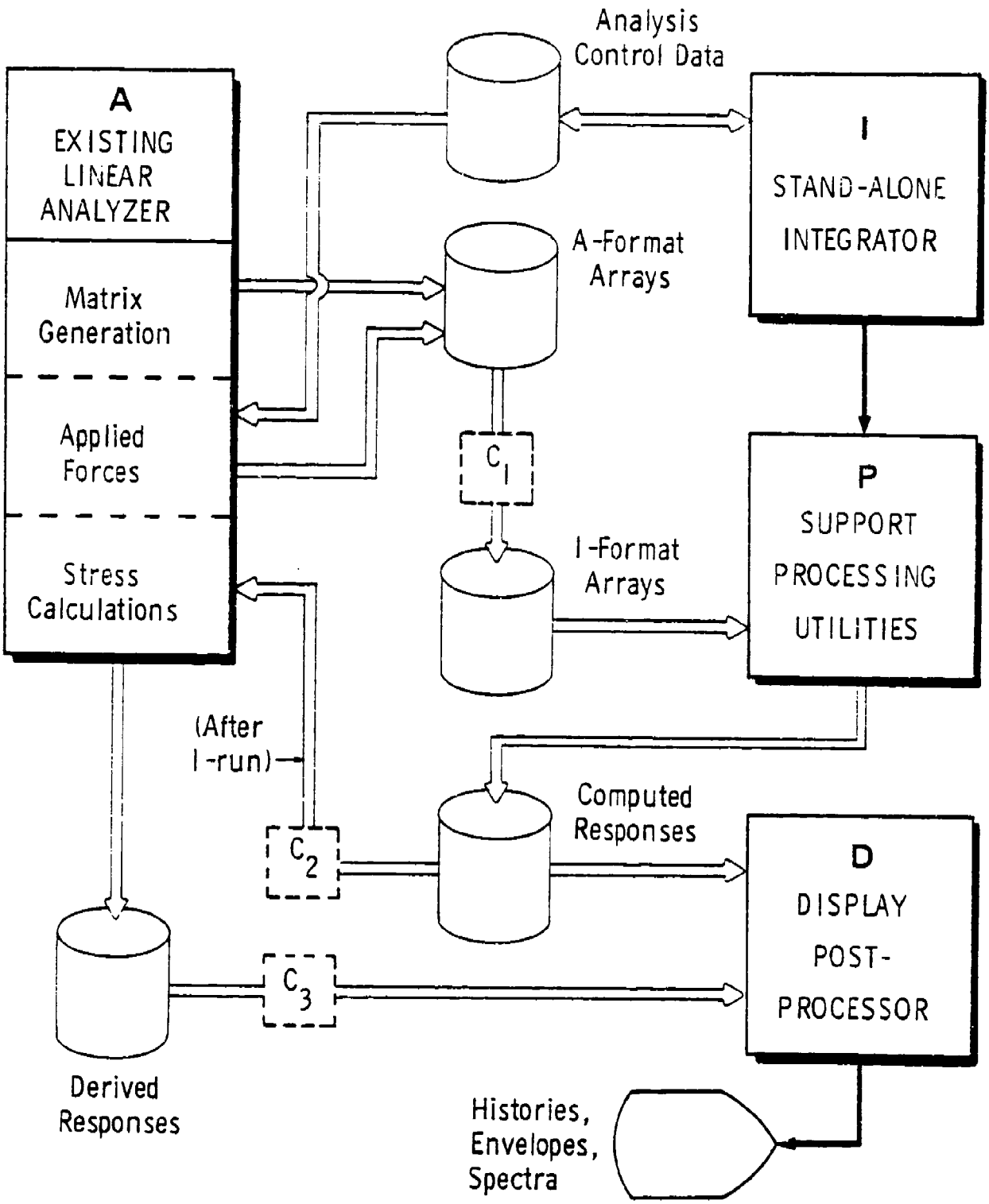


Fig. A-2 Operational Flow Diagram of Linear Dynamic Analysis with a Stand-Alone Integrator (→ Logic Control; ◊ Data Flow; C_i: Data Format Converters)

quantities such as strains and stresses is required, the simplest procedure is to send the computed responses back to the analyzer; derived responses can be then accessed by D. Two format converters, identified as C_2 and C_3 in Figure A-2, are generally required for this operation. Displayed results may take the form of time histories, envelope (peak response) plots, and response spectra.

An implementation of the operational flowchart in the form of data base-linked program network [40] is illustrated in Figure A-3. A key ingredient is a common data base manager (M) through which the global data base is accessed. In addition, computational modules such as the analyzer and the supporting matrix processors normally have their own local data bases (LDB), which contain intermediate result and auxiliary data structures. These local data bases are efficiently maintained in a "virtual memory" storage pool administered through local data base managers. These may be developer-supplied or part of the operating system library.

It is important to realize that the degree of modularity of this implementation is very high. For example, replacing an analyzer by another entails changing the converters (C_1, C_2, C_3), with no side effects propagating to the integrator. Conversely, incorporation of technical improvements in the integrator does not imply that the analyzers must be modified.

The ability to "plug in" different linear analyzers is particularly advantageous if segments of the structure are modeled by different programs. This is a common occurrence in large-scale engineering projects involving many subcontractors. The converter C_1 is then complemented by a "substructure-linking" module that ties the various models together.

Another advantage of the data base-linked organization is potentially smooth adaptability to distributed processing (segments of the problem may be assigned to different computers). For example, the display postprocessing stage can be most economically carried out on a minicomputer.

A.4 Transient Nonlinear Analysis

Figure A-5 depicts the operational flow for transient nonlinear analysis

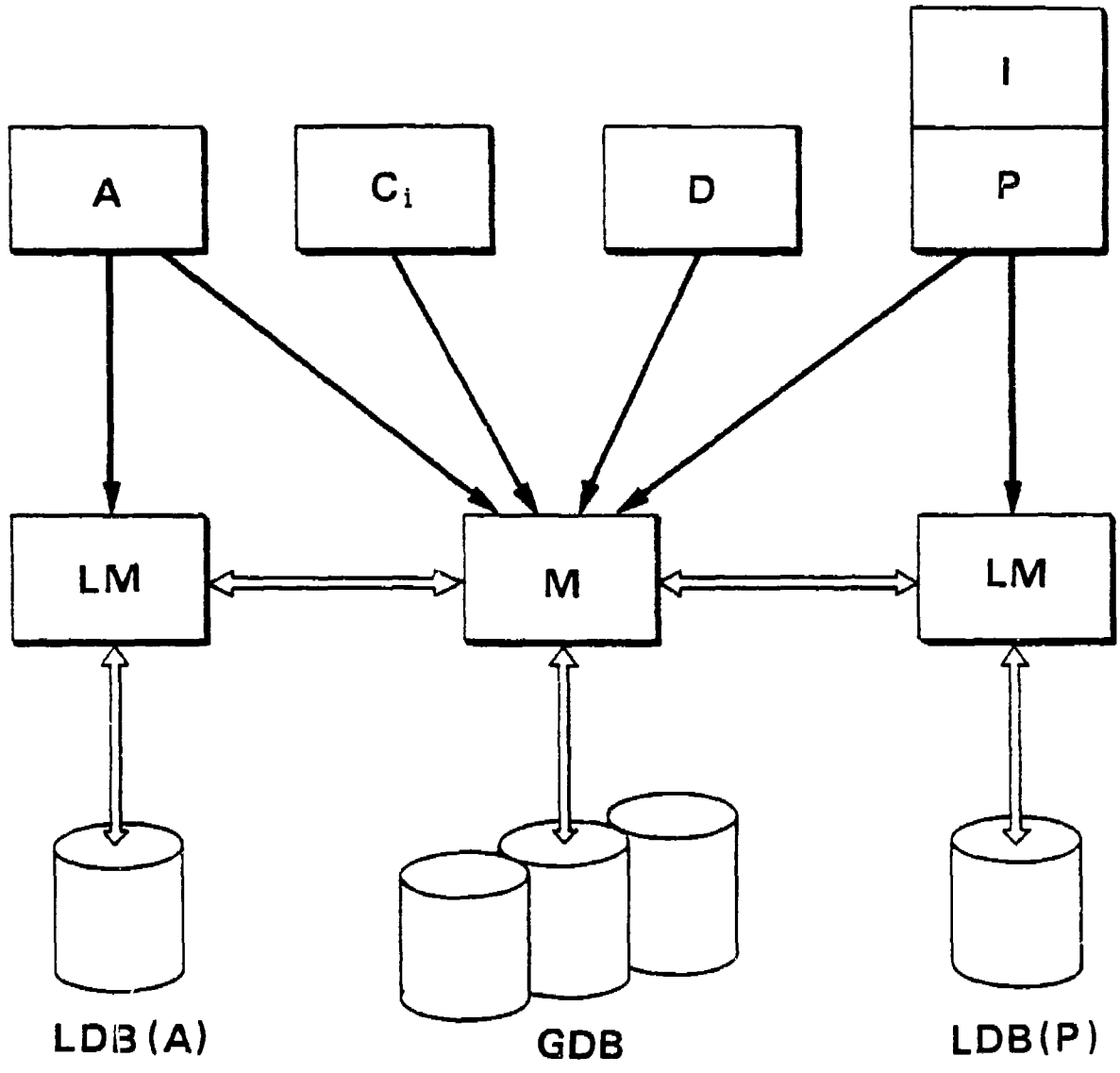


Fig. A-3 Implementation of Linear Dynamic Analysis Activities With a Database Management System (GDB, M = global database and its manager; LDB, LM = local database and its manager)

carried out through a pseudo-force based integrator. The key difference from the linear case is the occurrence of response feedback at each time step. This has a detrimental effect on the degree of modularity attainable as regards integrator-analyzer interaction.

Computed response information is used by the analyzer for evaluating stresses, nonlinear force terms and applied forces (if the latter are configuration-dependent). In problems involving material nonlinearities, the stress calculations also impacts the generation of stiffness matrices and of certain nonlinear force terms.

One of the constraints imposed by the lessened functional modularity is the elimination of the array format converter C_1. The array conversion would have to be exercised many times during the run, which can be expensive. The elimination of C_1 can be effected in several ways:

1. Embed C_1 in A so that arrays are directly produced in P-acceptable format.

2. Configure P so that various self-labeled array structures are accepted [42].

3. Do matrix processing in A so that most of the required array transmission is in the form of one-dimensional arrays.

In connection with the third approach, it is noteworthy to mention that explicit nonlinear integrators are inherently more modular than implicit integrators, inasmuch as all analyzer-to-integrator data can be transported in vector form, including the diagonal mass matrix $\underset{\sim}{M}$ (cf. Section 2.7).

How can the functional flowchart of Figure A-4 be implemented to maximize modularity? One possible solution is shown in Figure A-5. Here the analyzer and integrator operate as "coroutines" linked by an executive control component E.

Some operating systems require that E be a developer-supplied program, a restriction that forces A and I-P to be organized as parallel overlays.

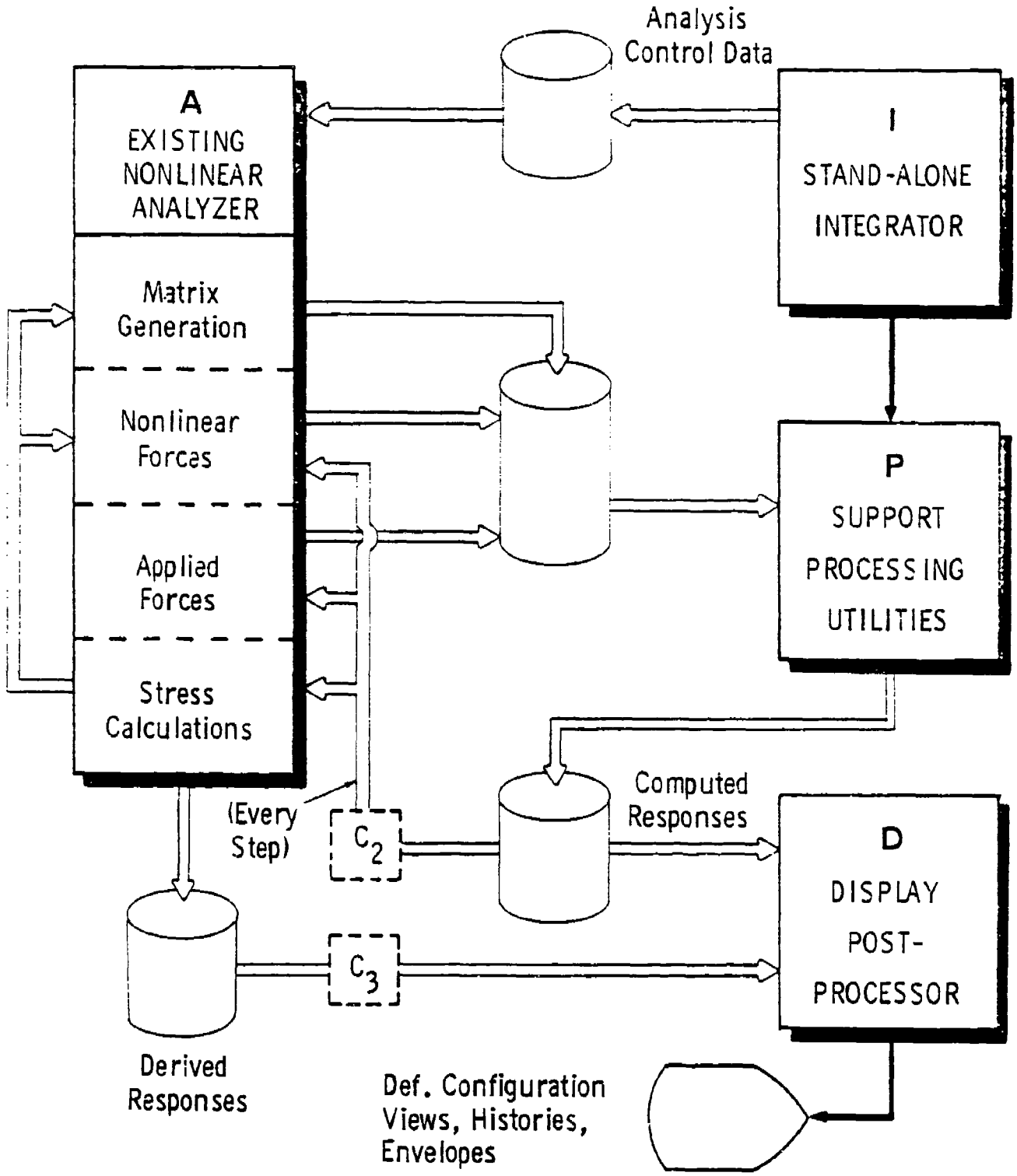


Fig. A-4 Operational Flow Diagram of Nonlinear Dynamic Analysis with a Pseudo-Force Based, Stand-Alone Integrator (→: Logic Control; ◇: Data Flow; C_i: Data Format Converter)

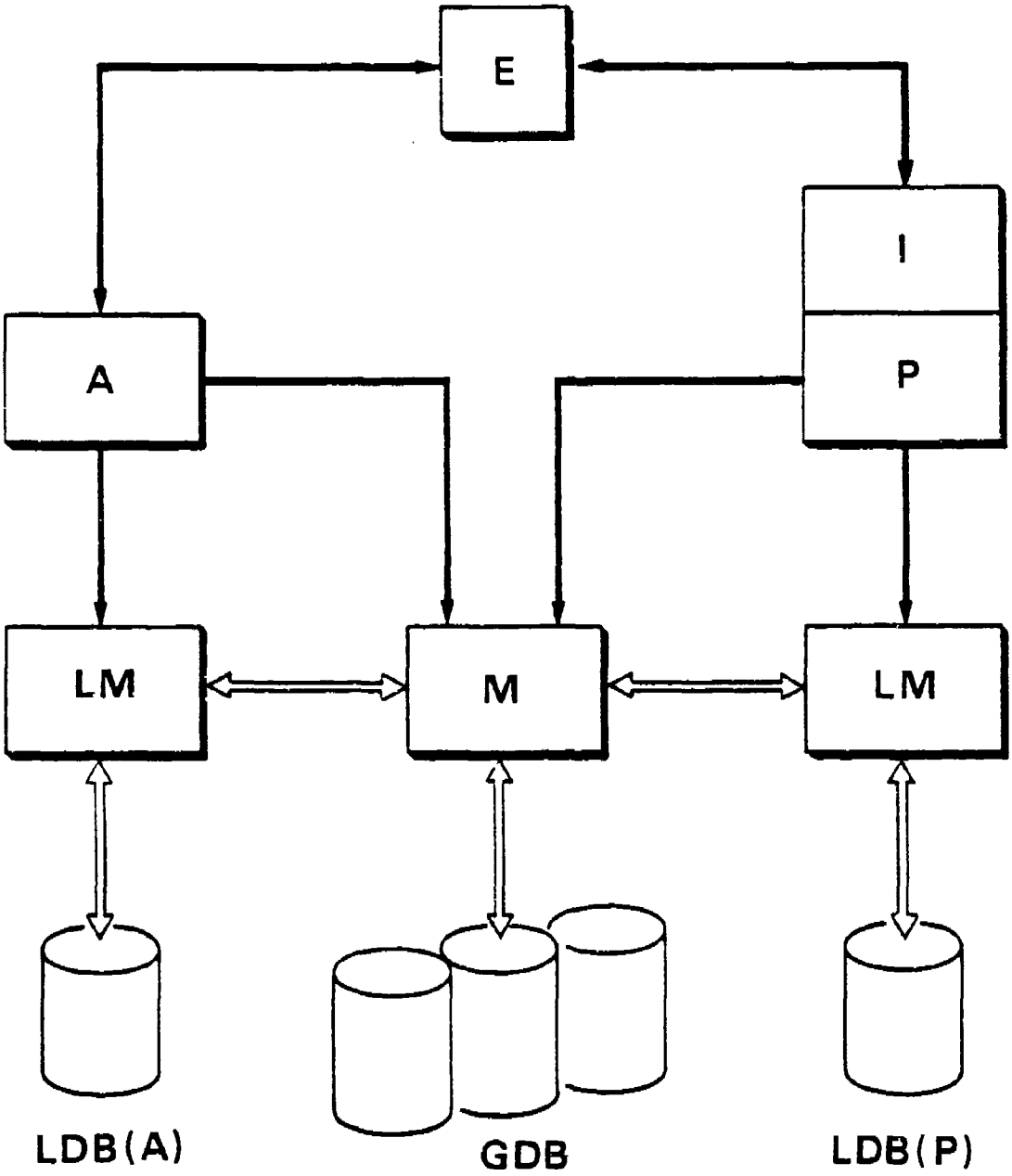


Fig. A-5 Implementation of Nonlinear Dynamic Analysis Activities With a Database Management System (cf. Fig. A-3) and an Execution Control Component (E). Display module not shown

This is unfortunate, for every modification of either program entails a complete reconstruction of the overlay image. More advanced operating systems allow E to be implemented as a catalogued control procedure that simply executes A or I according to run condition state variables. In this case, the two sides of Figure A-5 can be kept as physically independent programs served by a common version of the data base manager M.

A.5 The STINT Code

A stand-alone integrator called STINT has been developed following the loosely-coupled processor approach. Initial versions were primarily used as testbed of new techniques and reference source for subsequent development of production-level versions. STINT can operate as an implicit or explicit integrator.

The implicit integration branch uses the trapezoidal rule and the Park 3-step method [18] as basic formulas for treating linear and nonlinear problems, respectively. Processing of the latter relies upon a general pseudo-force approach used in conjunction with the matrix scaling technique described in Section 3. The (C1) computational sequence is presently used, although implementation of (J0) is anticipated. Further implementation details are given in Underwood and Park [5].

The explicit integration branch is based upon the variable-step central difference scheme [27]. Implementation details are given in Underwood and Park[33].

A long-term effort in "tuning up" problem-adaptive features of STINT, notably the automatic stepsize selection, has resulted in impressive but "staggered" performance gains in either the implicit or explicit branch. As a consequence, efficiency crossover points on the same test problem set have varied as new verions are prepared. Presently, the explicit integration branch is considered to be in superior shape.

Although STINT can switch from explicit to implicit mode, or vice-versa, during an analysis run, the switch points must be presently specified by the user. Our ultimate goal is to have STINT select the integration mode automatically according to a cost (merit) function minimization criterion;

this feature would greatly enhance the problem adaptivity of the integrator. The realization of this goal will have to wait, however, until stepsize control techniques in the implicit branch demonstrate the same degree of reliability currently achieved in the explicit branch.

ACKNOWLEDGMENT

The authors appreciate support of this work by the Office of Naval Research under Contract N00014-74-C-0355 and Lockheed Missiles and Space Company, Inc., under the Structural Dynamics Independent Research project.

REFERENCES

[1] C. A. Felippa and K. C. Park, Computational Aspects of Time Integration Procedures in Structural Dynamics, Part I: Implementation. To appear in J. Applied Mechanics, 1978.

[2] K. C. Park and C. A. Felippa, Computational Aspects of Time Integration Procedures in Structural Dynamics, Part II: Error Propagation. To appear in J. Applied Mechanics, 1978.

[3] C. A. Felippa, Procedures for Computer Analysis of Large Nonlinear Structural Systems, in Large Engineering Systems, ed. by A. Wexler (Pergamon Press, Oxford, 1977).

[4] J. A. Stricklin, J. E. Martinez, J. R. Tillerson, J. H. Hong, W. E. Haisler, Nonlinear Dynamic Analysis of Shells of Revolution by Matrix Displacement Method, Report 69-77, Texas A&M University, 1970.

[5] P. G. Underwood and K. C. Park, Implementation of a Variable-Step Integration Technique for Nonlinear Structural Dynamic Analysis, Trans. 4th Int. Conference on Structural Mechanics in Reactor Technology, ed. by T. A. Jaeger and B. A. Boley (SMIRT, San Francisco, 1977), paper M 5/2.

[6] P. S. Jensen, Transient Analysis of Structures by Stiffly Stable Methods, Computers & Structures 4 (1974) 615-626.

[7] P. Henrici, Discrete Variable Methods in Ordinary Differential Equations (Wiley, New York, 1962).

[8] C. W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations (Prentice-Hall, Englewood Cliffs, N.J., 1971).

[9] G. Dalquist and A. Björk, Numerical Methods (Prentice-Hall, Englewood Cliffs, N. J., 1975).

[10] N. M. Newmark, A Method of Computation for Structural Dynamics, J. Eng. Mech. Div., ASCE 85 (1959) 67-94.

[11] J. C. Houbolt, A Recurrence Matrix Solution for the Dynamic Response of an Elastic Aircraft, J. Aero. Sci. 17 (1950) 540-550.

[12] E. L. Wilson, L. Farhoomand, and K. J. Bathe, Nonlinear Dynamic Analysis of Complex Structures, Earthquake Eng. & Struct. Dyn. (1973) 241-252.

[13] K. C. Park, C. A. Felippa, and J. A. DeRuntz, Stabilization of Staggered Solution Procedures for Fluid-Structure Interaction Analysis, in Advances in Computational Methods for Fluid-Structure Interaction Problems, ed. by T. Belytscho and T. L. Geers, AMD-Vol. 26 (ASME, New York, 1977) 95-124.

[14] R. D. Krieg and S. W. Key, Transient Shell Response by Numerical Time Integration, in Advances in Computational Methods in Structural Mechanics and Design, ed. by J. T. Oden et al. (University of Alabama Press, Huntsville, 1972), 237-257.

[15] K. C. Park, An Efficient Implementation of Explicit Methods for Damped Second-Order Equations of Motion, Proc. Symposium on Applications of Computer Methods in Engineering, ed. by L. C. Wellford, Jr. (USC Press, Los Angeles, 1977) 635-646.

[16] A. Prothero and A. Robinson, On the Stability and Accuracy of One-Step Methods for Solving Stiff Systems of Ordinary Differential Equations, Math. Comp. 28 (1974), 145-162.

[17] A. R. Gourday, A Note on Trapezoidal Methods for the Solution of Initial Value Problems, Math. Comp. 24 (1970), 629-633.

[18] K. C. Park, An Improved Stiffly Stable Method for Direct Integration of Nonlinear Structural Dynamic Equations, J. Appl. Mech. 42 (1975), 464-470.

[19] K. C. Park, Evaluating Time Integration Methods for Nonlinear Dynamics Analysis, in Finite Element Analysis of Transient Nonlinear Structural Behavior (Ed. T. Belytschko et al.), ASME Applied Mechanics Symposia, Vol. AMD-14 (1975), 35-58.

[20] T. J. R. Hughes, Stability, Convergence, and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics, Comp. Struct. 6 (1976), 313-324.

[21] B. D. Westermo, The Solution of a Nonlinear Difference Equation Found in Numerical Analysis, California Institute of Technology, Rept. No. DYNL-78-01 (1978), Pasadena, Ca.

[22] K. C. Park, Practical Aspects of Numerical Time Integration, Comp. Struct. 7 (1977), 343-353.

[23] G. Dahlquist, Error Analysis for a Class of Methods for Stiff Nonlinear Initial Value Problems, in Lecture Notes in Mathematics, No. 506 (1976), 60-72.

[24] K. C. Park, A Variable-Step Implicit Method for Structural Dynamics Analysis, Part I: Theoretical Aspects, in preparation.

[25] K. C. Park and T. L. Geers, A Matrix Scaling Technique for Efficient Structural Transient Response Analysis, Proc. International Symposium on Innovative Numerical Analysis in Applied Engineering Science (1977), Versailles, France, 11-14.

[26] C. W. Gear and K. W. Tu, The Effect of Variable Mesh Size on the Stability of Multistep Methods, SIAM J. Numer. Anal. 11 (1974), 1025-1043.

[27] K. C. Park and P. G. Underwood, A Variable-Step Central Difference Method for Structural Dynamics Analysis, Part I: Theoretical Aspects, Lockheed Palo Alto Research Laboratory, Applied Mechanics Laboratory, Rept. No. LMSC-D626886 (1978).

[28] L. Euler, Opera omnia, series prima, Vol. 11 (Leipzig and Berlin, 1913).

[29] L. Euler, Opera omnia, series prima,,Vol. 12 (Leipzig and Berlin, 1914).

[30] R. A. Willoughby (Editor), Stiff Differential Systems (Plenum, New York, 1974).

[31] W. H. Enright and T. E. Hull, Test Results on Initial Value Methods for Non-stiff Ordinary Differential Equations, SIAM J. Numer. Anal. 13 (1976), 944-961.

[32] W. H. Enright, T. E. Hull, and B. Lindberg, Comparing Numerical Methods for Stiff Systems of O.D.E.'s, BIT 15 (1975), 10-48.

[33] P. G. Underwood and K. C. Park, A Variable-Step Central Difference Method for Structural Dynamics Analysis, Part II: Implementation and Performance Evaluation, Lockheed Palo Alto Research Laboratory, Applied Mechanics Laboratory, Rept. No. LMSC-D626886 (1978).

[34] P. J. Van Der Houwen, Construction of Integration Formula for Initial Value Problems (North-Holland, Amsterdam, 1977).

[35] B. Lindberg, A Stiff System Package Based on the Implicit Midpoint Method with Smoothing and Extrapolation in Stiff Differential Systems (A. Willoughby, Editor, Plenum, New York, 1974), 201-215.

[36] T. Belytschko and R. Mullen, Stability of Explicit-Implicit Mesh Partitions in Time Integration, submitted for publication (1978).

[37] T. J. R. Hughes and W. K. Liu, Implicit-Explicit Finite Elements in Transient Analysis; J. Appl. Mech. 45 (1978), 371-378.

[38] K. C. Park, Numerical Integration Techniques for Nonlinear Dynamic Analysis, presented at NASA Nonlinear Mechanics Workshop, Hampton, Va., 26-30 September 1977.

[39] K. C. Park, Semi-Explicit Methods for Transient Response Analysis of Structures, Lockheed Palo Alto Research Laboratory, Rept. No. LMSC/D623265 (1978).

[40] C. A. Felippa, Data Base Management in Scientific Computing, Part I: General Description. To appear in Computers & Structures.

[41] J. D. Aron, The Program Development Process (Addison-Wesley, Reading, Mass., 1974).

[42] P. S. Jensen, An Engineering Analysis System, to appear in Comm. ACM.